

Faculdade de Engenharia da Universidade do Porto



Navegação acústica passiva

Rui Filipe Pinho da Silva

VERSÃO PROVISÓRIA

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Major Automação

Orientador: Prof. Dr. Aníbal Castilho Coimbra de Matos

25 de Junho de 2017

Resumo

A presente dissertação, elaborada na sequência do mestrado em Engenharia Eletrotécnica e de Computadores, enquadrada nas atividades da Unidade Robótica e Sistemas Inteligentes do INESC TEC, visa a desenvolver um sistema capaz de localizar, em tempo real, um veículo submarino autónomo.

Para conseguir localizar a viatura, são utilizados marcos instalados em boias, com posições conhecidas pelo veículo. Estas boias são responsáveis pela emissão de sinais acústicos que são detetados pelo veículo, registando o tempo de chegada de cada sinal. Estes dados são utilizados para calcular a distância do veículo a cada marco, e através da triangulação destes valores é possível obter uma estimativa da posição do veículo.

A importância deste trabalho prende-se na forma de este apenas comunicar de forma passiva, o veículo será utilizado apenas como recetor. Esta propriedade permite reduzir a troca, pela metade, de sinais acústico entre o emissor e o recetor, e consequentemente, uma maior eficiência energética do veículo, visto que este é apenas como recetor.

A maior dificuldade deste sistema prende-se no facto de não existir sincronização de relógios. Esta característica traz muitas vantagens, assim como, a diminuição da complexidade do hardware utilizado e não existe necessidade de sincronizar todos os relógios no início de cada ciclo, no entanto, também torna o sistema mais complexo do ponto de vista do algoritmo matemático, e consequentemente, a complexidade computacional.

Neste documento será relatado o processo de desenvolvimento dos vários algoritmos utilizados neste sistema, que computadorizam todos os dados e estimam uma posição para o veículo.

A trajetória e o progresso do veículo serão apresentados em tempo real.

Abstract

This thesis, done in order of master degree in Electrical Engineering and Computers, inserted in the activities of the Robotics Unity and Intelligent Systems of INESC TEC, aims the development of a system capable of tracking and estimate the position, in real time, of an autonomous underwater vehicle.

To locate the AUV, its used points of reference installed in buoys, with known positions for the vehicle. This buoys are responsible to send acoustic signals that are detected by the AUV registering the time of arrival. This data is used to estimate the distance between the vehicle and the buoys, which is later to triangulate an estimated position of the vehicle.

The importance of this project is related is how this system communicates, that is passively, the vehicle only listen to the emitter, which are the buoys. This property reduces, by half, the exchange of signals between the emitter and the receptor, and consequently, a better energy efficiency, which is related to the fact that the AUV only listen to the buoys.

The greater difficulty of this systems, is associated to the fact of the inexistency of clock synchronism between the buoys and the AUV. This characteristic brings many benefices, which as lesser complexity of the hardware and there is no need to sync all the clocks in the system every time a cycle begins, in the other hand, the complexity of the mathematical algorithm increases, and therefore, the computation complexity as well.

In this document, will be related the process of the development of various algorithms crucial to the functioning of the system.

The trajectory and the progress of the vehicle will be presented in real time.

Agradecimentos

Quero começar por agradecer ao meu orientador, Aníbal Castilho Coimbra de Matos, por todo o tempo dedicado e interesse demonstrado no meu trabalho, sem a sua ajuda não teria conseguido chegar onde cheguei.

Quero agradecer aos meus pais e irmã, por todo o esforço e apoio que me foi dedicado, sem eles a realização desta dissertação não era possível.

Por último quero agradecer aos meus amigos e namorada por me ajudarem a desanuviar nesta altura, e por todos incentivos e dicas prestados.

Rui Silva

Conteúdo

Resumo	iv
Abstract.....	v
Agradecimentos	vii
Lista de figuras	xi
Lista de tabelas	xv
Abreviaturas e Símbolos	xvii
Capítulo 1	1
Introdução.....	1
1.1 - Contexto.....	1
1.2 - Motivação	2
1.3 - Objetivos	2
1.4 - Estrutura da dissertação.....	2
Capítulo 2	4
Revisão bibliográfica	4
2.1 - Métodos de navegação acústica	4
2.2 - Hardware utilizado	8
2.2.1 - AUV MARES	8
2.2.2 - Balizas acústicas	10
2.3 - Comunicação veículo - boias	12
2.3.1 - Tipos de comunicação	12
2.3.2 - Erros esperados	14
2.4 - Filtro de Kalman.....	15
2.5 - Técnica aplicada nesta dissertação	16
Capítulo 3	18
Algoritmo matemático	18
3.1 - Arquitetura da solução.....	18
3.2 - Desenvolvimento do algoritmo	18
3.2 - Limitações e suas soluções	22

Capítulo 4	25
Software desenvolvido.....	25
4.1 - Arquitetura da solução	25
4.2 - Mecanismos para evitar erros	26
4.3 - Testes ao algoritmo	38
Capítulo 5	34
Filtro de Kalman	34
5.1 - Desenvolvimento do filtro	34
5.2 - Testes ao filtro.....	35
Capítulo 6	41
Resultados	41
6.1 - Testes com diferentes posições dos beacons.....	42
6.2 - Testes com diferentes velocidades.....	44
6.3 - Testes com dados perdidos e/ou com ruído	47
6.4 - Testes com trajetórias complexas.....	54
Capítulo 7	60
Conclusão e trabalho futuro.....	60
5.1 - Satisfação dos objetivos.....	60
5.2 - Trabalho futuro	61
Referências	63

Lista de figuras

Figura 2.1 - Configuração típica de navegação acústica LBL [1]	5
Figura 2.2 - Representação esquemática de uma boia acústica [2]	5
Figura 2.3 - Triangulação genérica com três balizas	6
Figura 2.4 - Configuração típica da navegação SBL [1]	7
Figura 2.5 - Configuração típica da navegação USBL [1].....	7
Figura 2.6 - Técnica de OWTT [3]	8
Figura 2.7 - Técnica de TWTT [3]	8
Figura 2.8 - AUV MARES [3].	9
Figura 2.9 - Esquema do AUV MARES [3]	9
Figura 2.10 - Barramento PC104 [3]	10
Figura 2.11 - Portable Acoustic Navigation Buoys [4]	10
Figura 2.12 - Esquema mecânico da boia acústica [4]	11
Figura 2.13 - Ilustração do mecanismo “guarda-chuva” quando a boia entra na água [4]	11
Figura 2.14 - Diagrama da comunicação, entre sockets, recorrendo ao protocolo TCP.....	13
Figura 2.15 - Diagrama da comunicação, entre sockets, recorrendo ao protocolo UDP	14
Figura 2.16 - Fenómeno de multipath.....	15
 Figura 3.1 - Referencial cartesiano ortogonal com as posições das balizas e do veículo.	19
Figura 3.2 - Representação das distâncias entre as balizas e o AUV.	19
Figura 3.3 - Ilustração dos quatro quadrantes em um referencial cartesiano.	22

Figura 4.1 - Diagrama da comunicação, entre o AUV e uma boia acústica, implementada....	26
Figura 4.2 - Referencial usado nos testes	28
Figura 4.3 - Trajetória do veículo	29
Figura 4.4 - Diferença entre os instantes emitidos e recebidos.....	29
Figura 4.5 - Diferença entre a posição calculada e a real	30
Figura 4.5 - Diferença entre a posição calculada e a real, para o primeiro teste	30
Figura 4.5 - Diferença entre a posição calculada e a real, para o segundo teste	31
Figura 4.5 - Diferença entre a posição calculada e a real, para o terceiro teste.....	31
Figura 4.5 - Baseline do algoritmo utilizado.....	32
 Figura 5.1 - Trajetória realizada	 36
Figura 5.2 - Diferença entre a posição estimada pelo filtro de Kalman e a real.....	37
Figura 5.3 - Trajetória realizada com mudança de quadrante	37
Figura 5.4 - Diferença entre a os instantes emitidos e os recebidos.....	38
Figura 5.5 - Diferença entre as coordenadas estimadas pelo filtro de Kalman e as reais.....	38
Figura 5.6 - Diferença entre a posição calculada pelo algoritmo matemático e a real.....	39
 Figura 6.1 - Zona ótima de navegação.....	 41
Figura 6.2 - Referencial utilizado nos testes	42
Figura 6.3 - Caminho percorrido pelo veículo para testes de posição das balizas.....	42
Figura 6.4 -Diferença entre a posição calculada pelo algoritmo e a real	43
Figura 6.5 - Diferença entre a posição estimada pelo filtro e a real.....	43
Figura 6.4 - Diferença entre a posição calculada pelo algoritmo e a real	43
Figura 6.5 - Diferença entre a posição estimada pelo filtro e a real.....	43
Figura 6.6 - Diferença entre a posição calculada pelo algoritmo e a real	44
Figura 6.7 - Diferença entre a posição estimada pelo filtro e a real.....	44
Figura 6.8 - Trajetória para testes de velocidade	45
Figura 6.9 - Diferença entre a posição calculada pelo algoritmo e a real	45
Figura 6.10 - Diferença entre a posição estimada pelo filtro e a real	46
Figura 6.11 - Diferença entre a posição calculada pelo algoritmo e a real	46

Figura 6.12 - Diferença entre a posição estimada pelo filtro e a real.....	47
Figura 6.13 - Trajetória para testes com dados perdidos e/ou com ruído	48
Figura 6.14 - Diferença entre instantes recebidos e enviados, usados como controlo	48
Figura 6.15 - Diferença entre a posição calculada pelo algoritmo e a real, usado como controlo	49
Figura 6.16 - Diferença entre a posição estimada pelo filtro e a real, usado como controlo ..	49
Figura 6.17 - Diferença entre instantes recebidos e enviados, com perda de dados	50
Figura 6.18 - Diferença entre a posição calculada pelo algoritmo e a real, com perda de dados	50
Figura 6.19 - Diferença entre a posição estimada pelo filtro e a real, com perda de dados...	50
Figura 6.20 - Diferença entre instantes recebidos e enviados, com ruído nos dados.....	51
Figura 6.21 - Diferença entre a posição calculada pelo algoritmo e a real, com ruído nos dados	52
Figura 6.22 - Diferença entre a posição estimada pelo filtro e a real, com ruído nos dados ..	52
Figura 6.23 - Diferença entre instantes recebidos e enviados, com ruído e perda de dados ..	53
Figura 6.24 - Diferença entre a posição calculada pelo algoritmo e a real, com ruído e perda de dados	53
Figura 6.25 - Diferença entre a posição estimada pelo filtro e a real, com ruído e perda de dados	53
Figura 6.26 - Trajetória pretendida para o veículo	54
Figura 6.27 - Trajeto do veículo	55
Figura 6.28 - Diferença entre instantes recebidos e enviados, para ensaio do quadrado	55
Figura 6.29 - Diferença entre a posição calculada pelo algoritmo e a real, para ensaio do quadrado.....	56
Figura 6.30 - Diferença entre a posição estimada pelo filtro e a real, para ensaio do quadrado.....	56
Figura 6.31 - Trajeto pretendido do veículo	56
Figura 6.32 - Percurso percorrido pelo veículo	57
Figura 6.33 - Diferença entre instantes recebidos e enviados, para ensaio da cruz.....	57
Figura 6.34 - Diferença entre a posição calculada pelo algoritmo e a real, para ensaio da cruz	58
Figura 6.35 - Diferença entre a posição estimada pelo filtro e a real, para ensaio da cruz ...	58

Lista de tabelas

Tabela 4.1 - Dados recebidos pelo veículo	27
Tabela 4.2 - Dados recebidos pelo veículo com perda do beacon 3	27
Tabela 4.3 - Posição do veículo, com perda do beacon 3 do segundo conjunto de dados.....	27
Tabela 4.4 - Dados recebidos pelo veículo	27
Tabela 4.5 - Posição do veículo, com perda do beacon 3 do segundo conjunto de dados.....	27
Tabela 4.6 - Dados recebidos, do primeiro conjunto, pelo veículo	28
Tabela 4.7 - Dados recebidos pelo veículo, com perda de sinal no conjunto inicial	28

Abreviaturas e Símbolos

Lista de abreviaturas

ACK	<i>Acknowledge</i>
AUV	<i>Autonomous Underwater vehicle</i>
GPS	<i>Global Positioning System</i>
IP	<i>Internet protocol</i>
LBL	<i>Long Baseline</i>
OWTT	<i>One-way time travel</i>
SBL	<i>Short Baseline</i>
SPR	<i>Sinal por receber</i>
TCP	<i>Transmission Control Protocol</i>
TWTT	<i>Two-way time travel</i>
UDP	<i>User Datagram Protocol</i>
USBL	<i>Ultra short Baseline</i>

Capítulo 1

Introdução

Esta dissertação foi realizada no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto.

O presente capítulo é dedicado à introdução do problema, contendo o enquadramento a esta dissertação, seguido da motivação e os objetivos, sendo, por fim, apresentado uma estrutura do documento.

1.1 - Contexto

Nos sistemas robóticos terrestres, a navegação de veículos pode ser solucionada recorrendo à tecnologia GPS (*Global Positioning System*). No entanto, em sistemas subaquáticos o uso deste tipo de ferramenta é inadequado, uma vez que a água é um meio opaco para sinais eletromagnéticos, tornando soluções como o GPS ou ondas rádio inviáveis ao nosso sistema.

A navegação subaquática soluciona-se, sobretudo, com a determinação do posicionamento global, recorrendo à comunicação acústica. Esse processo é alcançado através de métodos bem conhecidos, nomeadamente, o que irá ser alvo de estudo neste trabalho, o Long Baseline (LBL). Outros métodos de navegação acústica irão ser abordados no capítulo 2.

Este método de navegação é conseguido com a utilização de equipamento externo que consiste num conjunto de balizas acústicas fixadas em boias no topo do oceano, sendo estas responsáveis pela emissão de sinais acústicos periodicamente que irão ser captados pelo veículo, desta forma é possível calcular a distância a que este se encontra a cada uma das balizas. Triangulando estas distâncias é possível conhecer a posição atual do veículo. A intervenção do operador neste processo é mínima e restringe-se à colocação e calibração da posição dos equipamentos necessários à localização.

1.2 - Motivação

A descoberta da acústica subaquática remonta ao tempo de Leonardo Da Vinci, quando este terá descoberto a possibilidade de ouvir navios à distância através de um tubo emergido no oceano.

Este tipo de comunicação pode ser encontrado em diversos ambientes como exploração científica e petrolífera, em missões de recolha de dados, pesquisa em ambientes difíceis e monitorização do ambiente marítimo minimizando a intervenção humana e por sua vez o risco de vidas humanas. Tipicamente estas comunicações são estabelecidas através de sinais acústicos, sendo este um enorme motivo para a sua exploração. Embora não seja a única forma de comunicação mantêm-se a mais eficiente e fiável até ao momento.

Atualmente os veículos subaquáticos têm uma grande importância quer a nível civil, quer a nível militar. Desta forma, enormes esforços têm sido desenvolvidos no sentido de os desenvolver, tornando-os cada vez mais autónomos e capazes de navegar sem ajuda.

1.3 - Objetivos

O objetivo deste trabalho centra-se no desenvolvimento de um sistema capaz de estimar, com alguma precisão, a posição de um AUV. De forma a cumprir este objetivo é necessário desenvolver um algoritmo em linguagem de programação. Este algoritmo deve ser robusto, suportando perdas e ecos dos sinais emitidos pelas boias, o que acontece com alguma frequência, e ainda facilmente adaptável para qualquer que seja a posição do veículo e das balizas.

Para estimar a posição do AUV, é imprescindível calcular expressões capazes de, através dos tempos de chegada dos sinais acústicos, a determinar de uma forma precisa.

Para ser possível observar o comportamento e movimento do veículo, pelo operador, em tempo real, é imprescindível o desenvolvimento de uma plataforma responsável pela representação destes dados de forma explícita.

1.4 - Estrutura da dissertação

Para além deste capítulo, existem ainda mais seis capítulos. No capítulo 2 é feita a revisão bibliográfica e são apresentadas as soluções atuais para a navegação de veículos submarinos autónomos. No capítulo 3 é feita uma análise do algoritmo matemático essencial à estimação da posição do AUV, explicando como foi obtido e quais as suas limitações e características. No capítulo 4 é analisado o algoritmo C, é feita uma pequena introdução teórica sobre UDP e TCP, seguido por uma análise sobre qual a arquitetura selecionada para a elaboração deste trabalho e ainda quais as características do algoritmo, e quais os métodos optados para combater os erros mais comuns. No capítulo 5 é apresentado o desenvolvimento de um filtro de Kalman, utilizado para estimar a posição do veículo. No capítulo 6 é relatado a estrutura do sistema, assim como, vários exemplos do resultado final do trabalho. Finalmente, no capítulo 7 são retiradas as conclusões acerca da satisfação dos objetivos propostos e feitas considerações sobre possíveis desenvolvimentos futuros.

Capítulo 2

Revisão bibliográfica

O objetivo deste capítulo é fazer uma análise concisa e crítica do estado de arte inerente ao tema desta dissertação.

São introduzidos alguns conceitos essenciais à compreensão do tema em estudo, tal como os métodos de navegação acústica de veículos submarinos autônomos (AUV), e com mais detalhe o que irá ser desenvolvido neste caso de estudo.

De seguida irão ser abordados métodos para o seguimento em tempo real do AUV, e com maior pormenor o que irá ser utilizado nesta dissertação.

Por fim irei demonstrar o sistema utilizado neste projeto, e quais as suas propriedades.

2.1. Métodos de navegação acústica

Existem diversos métodos para a navegação de AUV's, distinguindo-se principalmente pelo número de balizas acústicas utilizadas e pela forma de medir a distância ao veículo, que pode ser através de diferenças temporais ou através do ângulo dos sinais acústicos.

O método de LBL (*Long Baseline*), representado na figura 2.1, utiliza uma rede de balizas acústicas, no mínimo três destes sensores para ser possível obter um resultado correto, normalmente instaladas em boias e fixadas em solo subaquático que funcionam como pontos de referência para o veículo. Estas boias são fixadas ao longo de uma área em que se pretende navegar o AUV, a qual é capaz atingir grandes distâncias (100 m a +6000 m) com um erro associado inferior a 1m que é independente da profundidade da operação.

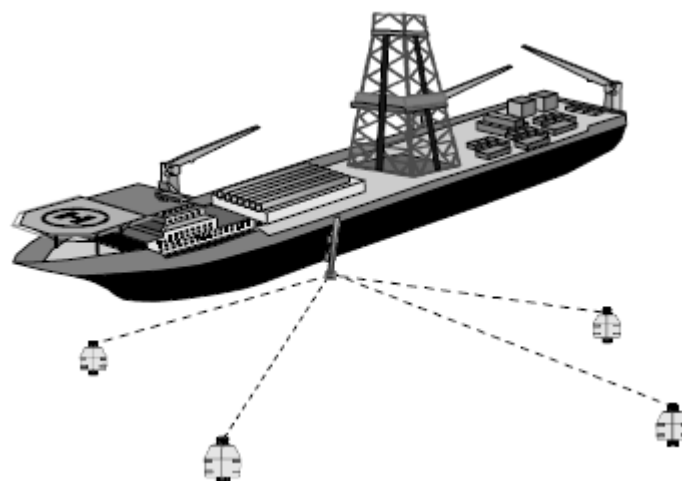


Figura 2.1 - Configuração típica de navegação acústica LBL [1].

As balizas acústicas têm na sua composição um dispositivo eletrônico chamado *transponder* acústico, que é responsável por receber e retransmitir um sinal acústico.

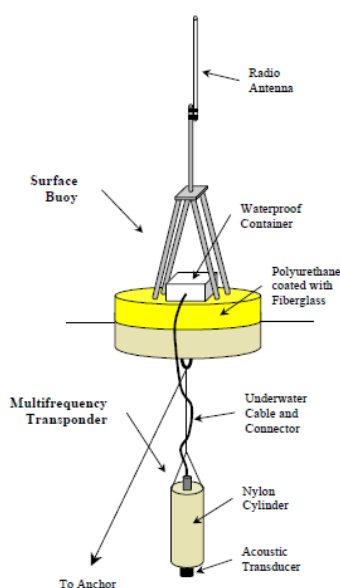


Figura 2.2 - Representação esquemática de uma boia acústica [2].

Estas boias, representadas na figura 2.2, são responsáveis emitir um sinal acústico com uma frequência única para que seja possível o AUV entender com qual baliza está a comunicar. Através do tempo de voo dos vários sinais recebidos é possível obter a distância, através da expressão matemática da velocidade, a que se encontra o veículo a cada um dos marcos. Após este processo é possível triangular as distâncias calculadas de forma a obter uma localização do AUV, como se pode ver na figura 2.3.

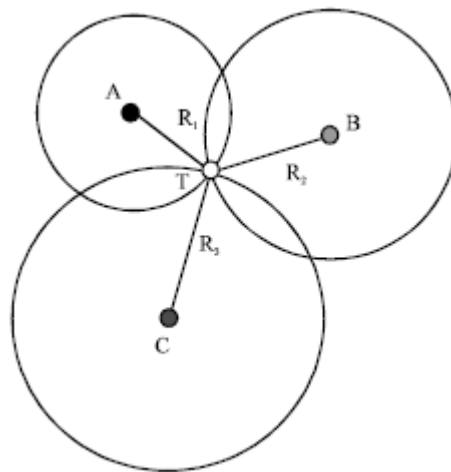


Figura 2.3 - Triangulação genérica com três balizas.

A figura 2.3 é um esquema conceptual do método de posicionamento acústico em duas dimensões. Cada círculo representa a distância de cada marco ao veículo, onde a interseção é a localização do AUV. É possível verificar que com apenas três marcos é possível triangular corretamente a posição de um veículo, dado que a profundidade é genericamente conhecida e calculada através de sensores de pressão, logo não é estritamente necessário recorrer a um quarto marco.

Um sistema muito utilizado atualmente funciona com frequências entre 18 – 36 kHz com um alcance entre 2 – 3.5 km [1], e ainda com uma profundidade máxima de 3.5 km, após esta distância o veículo não consegue “escutar” os marcos.

Neste tipo de aplicações existem algumas fontes de erro, tais como:

- As características da água, as quais fazem variar a velocidade a que o sinal acústico se propaga. Este tipo de erro é irrelevante para pequenas distâncias, no entanto para distâncias de alguns quilómetros, o caso muda de figura, tornando-se inadmissíveis para o sistema. De forma a reduzir estes erros é necessário medir de forma minuciosa a velocidade a que o som no meio em que o veículo se encontra.
- O *drift* associado ao relógio, esta variação afeta a precisão da medição do tempo de voo. Este problema pode ser resolvido recorrendo a métodos em que não seja necessária a sincronização de relógios.
- A calibração das boias acústicas. Está associado à posição dos marcos que podem variar no tempo com a ondulação do meio, é necessária uma calibração inicial extremamente precisa e verificar regularmente se a sua posição não varia, de forma a diminuir este tipo de erros.

Outro tipo de navegação é SBL (*Short Baseline*) representado na figura 2.4, neste método é necessária apenas um marco o qual não necessita de ser ancorado ao fundo do oceano com uma área de operação na ordem dos 20-50 m. O veículo contém um conjunto de *transponders* que emitem um sinal com diferentes frequências. O cálculo da posição do veículo é similar ao método de LBL, que consiste em obter os intervalos de tempo que cada sinal acústico leva a chegar ao marco, os quais irão ser posteriormente utilizados para calcular qual a distância à baliza acústica e, por fim, triangular estas distâncias, de forma a obter a localização da embarcação.

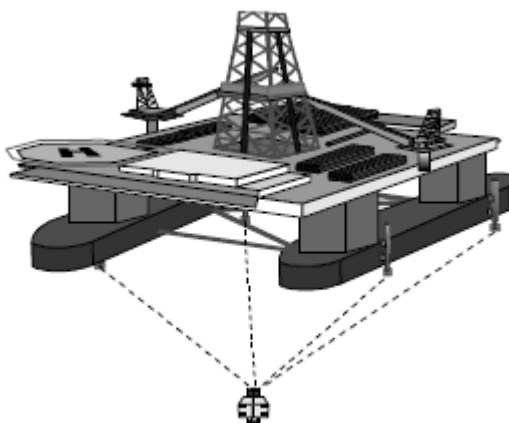


Figura 2.4 - Configuração típica da navegação SBL [1].

Existe ainda outro método, representado na figura 2.5, designado por USBL (*Ultra Short Baseline*), que consiste em medir o ângulo em que difere o sinal emitido e o recebido, através desta diferença é possível calcular a distância ao veículo. Este método requer apenas a utilização de um *transponder* e de uma boia acústica e consegue medir distância até 10 cm.

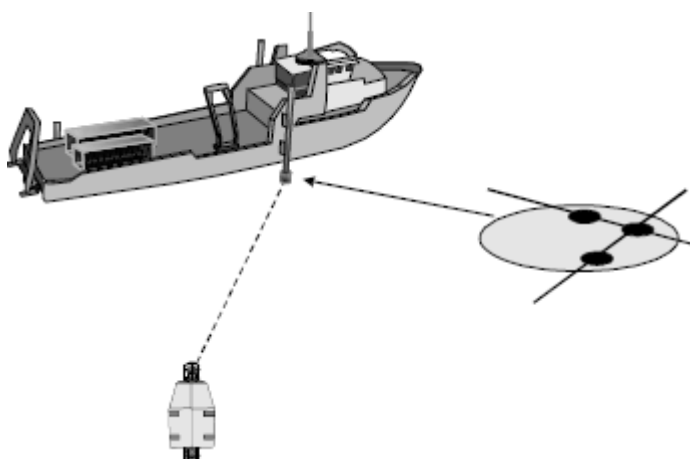


Figura 2.5 - Configuração típica da navegação USBL [1].

Nas duas primeiras técnicas anteriores, LBL e SBL, podemos optar por um sistema com sincronização de relógio, que consiste em termos o recetor e o emissor com relógios sempre sincronizados, desta forma, é possível calcular o tempo de voo dos sinais acústicos de forma mais simples, sabendo qual o instante a que foi emitido o sinal e quando o mesmo foi recebido. No entanto, como já tinha dito anteriormente, esta técnica acarreta mais erros devido ao *drift*, ou seja, deriva temporal que os relógios apresentam. Este problema pode ser solucionado com a aquisição de instrumentos com *drift* muito reduzido ou recorrer a técnicas a que não existe sincronização de relógio, ou seja, não é conhecido o instante a que foi emitido o sinal, apenas quando foi recebido, tal como o caso de estudo desta dissertação.

Em técnicas com sincronização de relógio, tipicamente, é usada uma configuração conhecida como *One-Way Time Travel* (OWTT), representada na figura 2.6, onde ToF representa o tempo de voo do sinal, t_f o tempo de aquisição do sinal, B1 uma boia acústica e B2 o AUV.

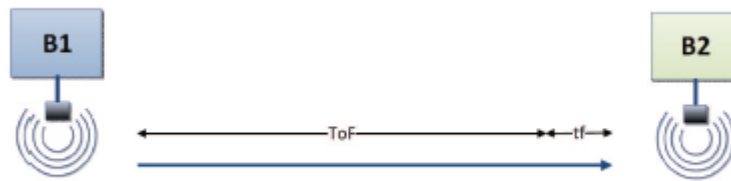


Figura 2.6 - Técnica de OWTT [3].

Existe ainda a configuração de *Two-Way Time Travel* (TWTT), representada na figura 2.7, onde tp e td é o tempo de processamento e reenvio do sinal, que geralmente, é empregue em sistema sem sincronização de relógio. Desta forma mesmo sem a sincronização a boia acústica conhece o instante em que o sinal foi emitido e recebido. No entanto apresenta algumas desvantagens, tais como:

- Maior consumo de energia pelo AUV, levando a sua bateria se esgote mais rapidamente.
- Maior número de sinais acústicos trocados entre as balizas e o veículo, aumentando desta forma o risco de perda de sinais.
- Aumento do tempo de cada ciclo para o dobro face à técnica de OWTT.



Figura 2.7 - Técnica de TWTT [3].

2.2. Hardware utilizado

2.2.1. AUV MARES

O AUV MARES (*Modular Autonomous Robot for Environment Sampling*), representado na figura 2.8, é um veículo submarino autónomo, desenvolvido pelo grupo *The Ocean Systems Group* (Oceansys) da Faculdade de Engenharia da Universidade do Porto juntamente com o ISR-Porto (instituto de sistemas robóticos - Porto), com 1.5 m de comprimento e 20 cm de diâmetro, atingindo velocidades horizontais na ordem dos 2 m/s, sendo capaz de mergulhar até 100 m de profundidade. Este veículo pode ser configurado para carregar uma vasta variedade de sensores, recolhendo informação do meio em que navega enquanto se movimenta por rotas predefinidas.



Figura 2.8 - AUV MARES [4].

As suas partes mecânicas foram desenhadas usando Solidworks® CAD software e construídas com poliacetal, com pequenas partes em aço inoxidável e alumínio.

Na parte central do veículo encontra-se uma secção à prova de água, onde se encontram todos os componentes eletrónicos necessários ao funcionamento do AUV. As restantes secções são permeáveis. O que permite, de uma forma simplificada, trocar de sensores ou até mesmo os reposicionar.

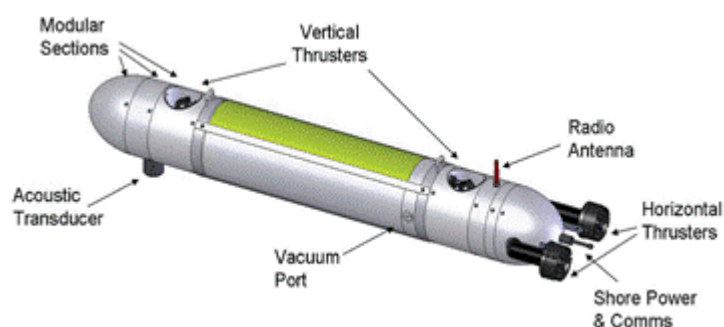


Figura 2.9 - Esquema do AUV MARES [4].

O veículo assemelha-se a um torpedo, com elipsoides em ambas as extremidades, diminuindo assim o atrito causado pelo seu movimento.

Ao contrário dos outros pequenos AUV's que usam lemes de inclinação para alterar a sua posição vertical, o MARES contém 2 propulsores verticais que permitem alterar a posição vertical sem alterar a posição horizontal. Isto é uma vantagem no caso de missões de varrimento sonar ou filmagem subaquática, nas quais um AUV com lemes de inclinação terá que alterar a sua inclinação para acompanhar os desníveis do terreno, comprometendo os dados, algo que não ocorre no MARES.

A maior parte da energia gasta pelo AUV é devido à propulsão, com apenas uma pequena parte gasta pelo sistema computacional. No MARES, toda a energia é armazenada numa bateria recarregável de iões de lítio, capaz de fornecer 600 Wh com uma tensão de 14.4 V.

A duração da bateria depende bastante da velocidade do veículo, quer no plano horizontal, quer no plano vertical. No caso de missões tipicamente horizontais, com poucas alterações na profundidade, o veículo tem energia para cerca de 8 - 10 h de operação, cerca de 40 Km.

Embora estes valores sejam relativamente baixos, são suficientes para a maior parte das missões.

O sistema computacional é baseado num barramento PC104, representado na figura 2.10, composto por uma placa de alimentação, uma placa de processador e placas de interface com os periféricos. De forma a guardar os dados recolhidos em cada missão é usado um *flash disc*.



Figura 2.10 - Barramento PC104 [4].

O software principal do veículo foi desenvolvido em C e corre num kernel Linux, sendo composto por um conjunto de processos independentes. Desta forma, não só a modularidade e robustez do sistema são aumentadas, mas o seu debug e recuperação de erros inesperados tornam-se mais simples. A comunicação entre os módulos é baseada no mecanismo de passagem de mensagens e usa protocolo UDP.

2.2.2. Balizas acústicas

Um dos fatores mais importantes na operação de veículos autónomos é a computação, em tempo real, de uma estimativa precisa da posição do veículo. Esta estimativa é fundamental para o sucesso da missão. Para isso o INESC TEC desenvolveu um novo tipo de boias acústicas, denominadas de *Portable Acoustic Navigation Buoys*, representadas na figura 2.11.



Figura 2.11 - Portable Acoustic Navigation Buoys [5].

Estas boias são constituídas por um cilindro de polycarbonato à prova de água, com um diâmetro de 12 cm, onde se encontra todo o equipamento eletrónico.

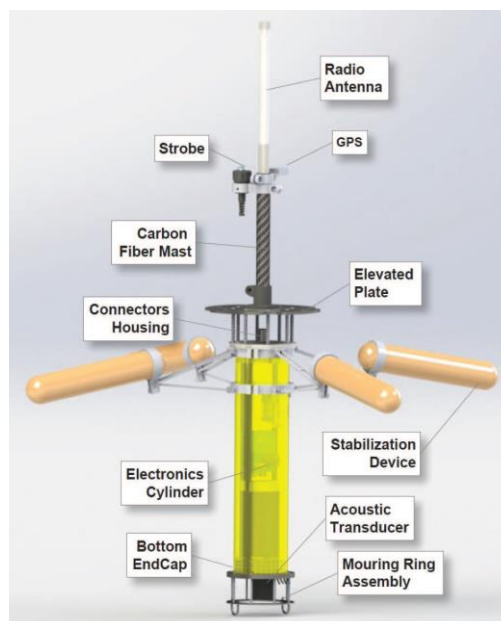


Figura 2.12 - Esquema mecânico da boia acústica [5].

Todos os componentes foram depositados de tal forma de forma a que o cilindro fique estável na vertical, representado na figura 2.12. Quando é depositado a parte mais em baixo fica submersa enquanto que a parte superior fica sobre a água. Na parte superior encontra-se uma antena de rádio e GPS e uma luz de sinalização. Na zona inferior está o transdutor protegido por uma caixa metálica, onde a ancora se segura.

Para ter uma maior estabilidade, um dispositivo que faz lembrar um guarda chuva, foi implementado na boia. Quando esta entra na água este sistema estende quatro estabilizadores flutuantes ficando com um diâmetro de 116 cm quando estendido e 35 cm quando dobrado, na figura 2.13 encontra-se representado este processo. Além de garantir uma maior estabilidade também fornece uma maior flutuação à boia.



Figura 2.13 - Ilustração do mecanismo “guarda-chuva” quando a boia entra na água [5].

Dentro do cilindro, está uma *open frame* responsável por estabelecer a ligação entre o utilizador e todas as propriedades da boia, assim como os seus sensores. Mecanicamente este dispositivo é capaz de suportar um recetor de GPS e sinal de rádio. Existe ainda espaço para instalar um microprocessador, assim como o Odroid, BeagleBone ou Raspberry Pi. Instalado

diretamente a uma porta USB está uma *dongle* responsável por estabelecer uma ligação Wifi que consegue comunicar até 200 m. Para comunicações de distância ainda maiores é usado um sinal rádio, juntamente com uma antena que conseguindo atingir distâncias superiores a 1 Km, mesmo quando temos apenas 9600 Baud. Para se conseguir obter um sinal GPS é utilizado um GPS *addon*, ligado diretamente a um *Ublox Neo6 chip*, que posteriormente está conectado ao microprocessador. Este dispositivo é capaz de fornecer uma taxa de atualização de 5 Hz e um sinal PPS (*Pulse Per Second*) que é utilizado em maior parte das comunicações acústicas. Suporta todo o tipo de comunicação *Baseline*, assim como LBL, usando TWTT ou OWTT, assim como descrito anteriormente. Para medir a estabilidade do sistema foi instalado uma bússola digital.

Em termos energético, esta boia consome em média 6 W e quando utilizada como transponder com uma frequência de 1 Hz consome mais 0,8 W. Para alimentar a boia é utilizada uma bateria recarregável de Li-Po capaz de fornecer 185 Wh. Em testes feitos anteriormente, a duração da bateria foi de, aproximadamente:

- 30 h, se apenas estiver a receber sinais.
- 28 h, em comunicações OWTT com uma frequência de 0,5 Hz.
- 26 h, em comunicações OWTT com uma frequência de 1 Hz.

Em relação ao software, é utilizado um PC embutido onde corre um sistema de Linux e a comunicação entre os módulos é baseada no mecanismo de passagem de mensagens e usa protocolo UDP. Os dados de cada missão podem ser facilmente exportados para múltiplos formatos, incluindo Matlab.

2.3. Comunicação veículo - boias

2.3.1. Tipos de comunicação

Hoje em dia, graças à adoção, como *standard*, do protocolo IP, os dispositivos conseguem comunicar facilmente entre si. Existem várias tecnologias que permitem a comunicação entre máquinas. Atualmente, um dos mecanismos mais utilizados para este efeito são os *sockets*. Esta tecnologia, normalmente, é utilizada para implementar um modelo cliente/servidor, que permite a comunicação entre o cliente e o servidor.

Existem principalmente dois tipos de *sockets*, os *Stream Sockets* e os *Datagram Sockets*, também designados de SOCK_STREAM e SOCK_DGRAM, respetivamente.

Os *Stream Sockets* são conhecidos como *sockets* com uma conexão orientada, e utilizam maioritariamente TCP como protocolo de comunicação. Este tipo de *sockets* enviam toda a informação pela ordem inicialmente estabelecida, ou seja, se se enviar duas mensagens na ordem “1,2”, estas mensagens chegam na mesma ordem, “1,2”, não existindo erros nem perdas dos pacotes de informação enviados.

Os *Datagram Sockets* são conhecidos como *sockets* sem conexão, ou seja, não necessitam de permissão do servidor para se conectarem a ele, e desta forma, enviar mensagens. Este tipo de *sockets* utiliza um protocolo de comunicação conhecido como UDP, que ao contrário do TCP, podem conter erros e não garantem que a informação chega ao seu destino, o que pode ser solucionado com uma mensagem de ACK (*acknowledge*), por parte do servidor, assim que receba a informação enviada pelo cliente.

Na figura 2.14, encontra-se um exemplo da comunicação entre *sockets* usando o protocolo TCP.

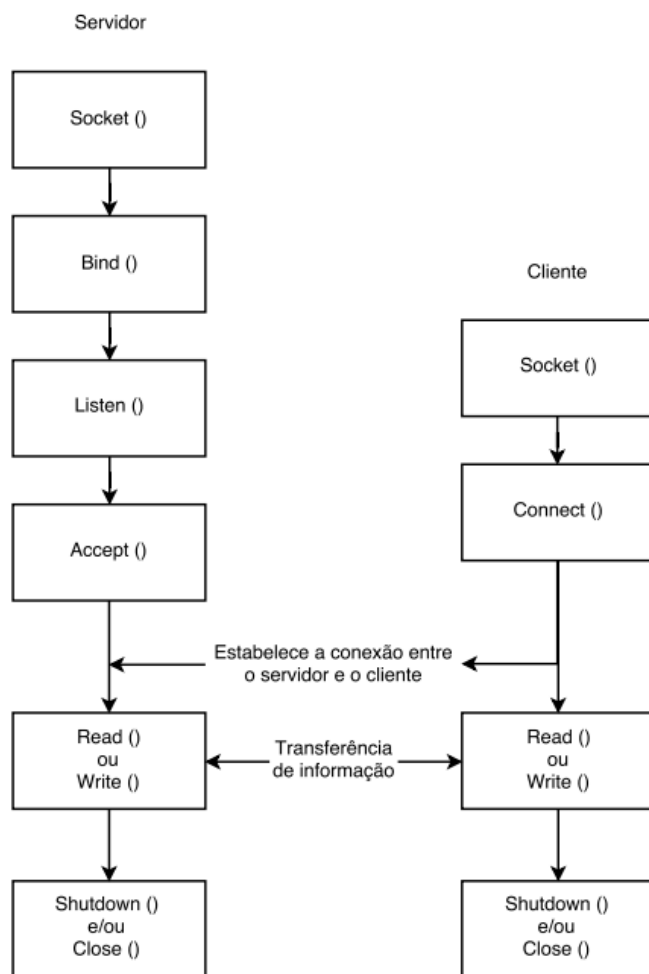


Figura 2.14 - Diagrama da comunicação, entre *sockets*, recorrendo ao protocolo TCP.

Na figura 2.15, encontra-se um exemplo da comunicação entre *sockets* usando o protocolo UDP.

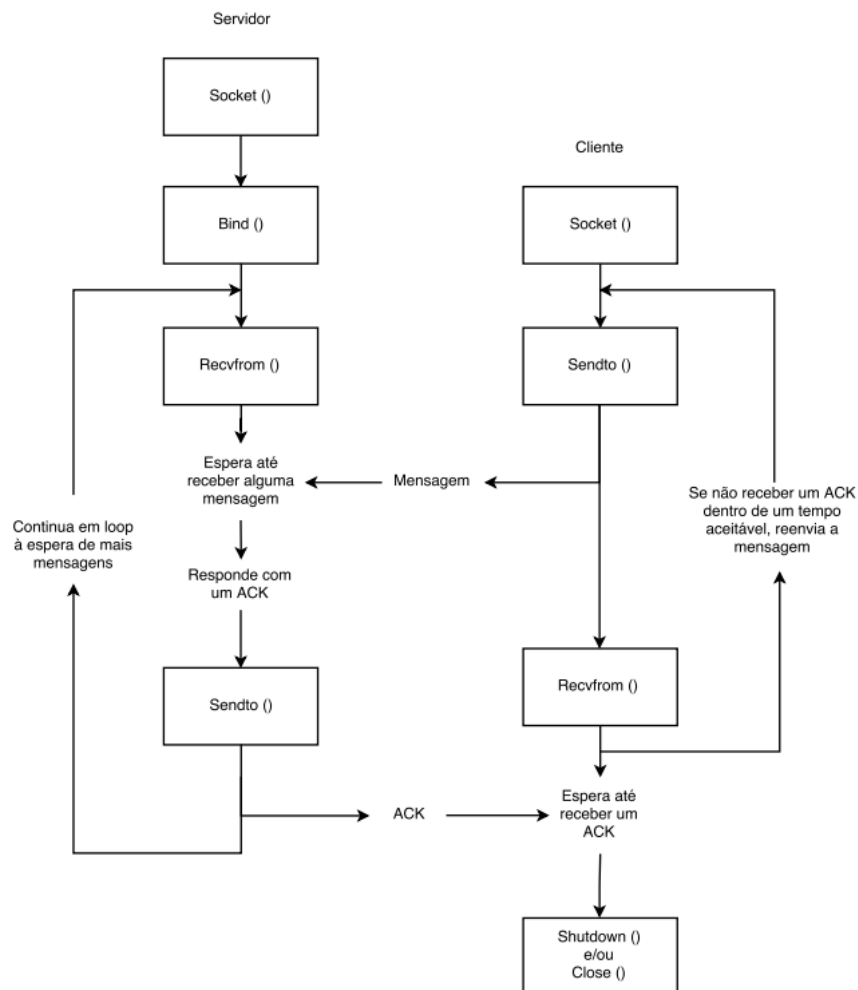


Figura 2.15 - Diagrama da comunicação, entre *sockets*, usando o protocolo UDP.

Como se pode verificar nas imagens 2.14 e 2.15 a maior diferença entre o protocolo TCP e o UDP, reside, em que no protocolo UDP não é necessário o cliente se conectar e aguardar a aceitação por parte do servidor para enviar pacotes de dados.

2.3.2. Erros esperados

Num sistema de navegação acústica existem sempre inúmeros erros associados, mas muitos deles podem ser ultrapassados, assim como a reflexão e perda de sinais.

A reflexão de sinais, também conhecido como *multipath*, representado na figura 4.4, consiste, como o próprio nome indica na reflexão do sinal enviado pela boia acústica na superfície ou no fundo do oceano ou ainda em algum corpo presente no meio.

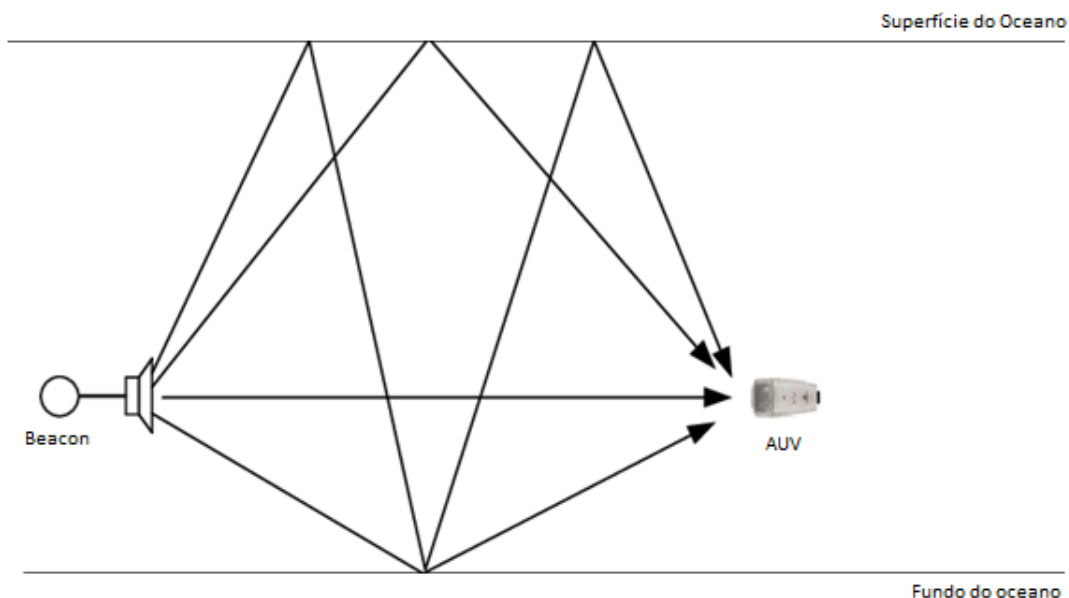


Figura 2.16 - Fenómeno de multipath.

Este fenômeno leva a que o AUV possa receber mais do que uma vez o mesmo sinal, provocando incoerências na aquisição dos sinais. De forma a combater este problema, foi implementado um mecanismo no algoritmo capaz de perceber se o sinal foi ou não refletido, tendo em conta o instante a que chegou o sinal anterior da mesma boia, e o instante a que chegou o atual.

Em forma de exemplo, vamos supor que o AUV recebeu um sinal, e reconheceu que este pertence ao *beacon* 1 e chegou no instante 0,01 segundos. Passado um curto período de tempo o veículo detetou outro sinal proveniente do *beacon* 1, o qual chegou no instante 0,20 segundos. Tendo em conta que cada boia acústica emite com uma frequência de 1 Hz, ou seja, apenas emite um sinal de cada 1 s em 1 s, é possível verificar que este segundo sinal percorreu um percurso que, não foi o direto, e chegou uns segundos mais tarde, devendo ser considerado um sinal refletido e por este motivo não ser sequer utilizado no sistema.

Imaginemos agora que o segundo sinal chegava no instante 1,03 segundos, tendo em conta o instante anterior, de 0,01 segundos, verifica-se que a diferença entre os dois é superior a 1 segundo. Posto isto, podemos considerar que o segundo sinal pertence à geração posterior ao primeiro sinal.

Quando este fenômeno acontece, o sinal refletido, simplesmente, é ignorado, e não é gravado no algoritmo C nem no ficheiro de texto.

A perda de sinais, já é algo mais problemático, este fenômeno pode acontecer devido à atenuação do sinal enviado, que quando chega ao recetor não possui frequência suficiente para ser detetado, ou quando existe um obstáculo no caminho e o sinal acaba por ter de percorrer uma distância superior, em relação ao caminho direto, o que provoca uma elevada atenuação. Podem ainda existir ocasiões onde simplesmente existe perda do pacote de dados, o que acontece com alguma frequência em comunicações através de UDP, como mencionado anteriormente.

2.4. Filtro de Kalman

O filtro de Kalman é um método matemático criado por Rudolf Kalman. O seu propósito é, utilizando medições de grandezas realizadas ao longo do tempo, contaminadas com ruído e outras incertezas, gerar resultados que tendam a se aproximar dos valores reais das grandezas medidas e valores associados.

O filtro de Kalman produz estimativas dos valores reais de grandezas medidas e valores associados predizendo um valor, estimando a incerteza do valor predito e calculando uma média ponderada entre o valor predito e o valor medido. O peso maior é dado ao valor de menor incerteza, portanto será quanto menor a incerteza mais o filtro confia que os valores apresentam uma menor quantidade de erros. As estimativas geradas pelo método tendem a estar mais próximas dos valores reais do que as medidas originais pois, a média ponderada apresenta uma melhor estimativa de incerteza que ambos os valores utilizados no seu cálculo.

O filtro de Kalman é um eficiente filtro recursivo, isto significa que apenas a estimativa do estado no passo anterior e a medição atual são necessários para computar a estimativa do estado atual.

Este filtro utiliza um modelo dinâmico de um sistema (as equações do movimento, por exemplo), entradas de controlo conhecidas (como a velocidade) e medições (como os instantes escutados pelo AUV) para gerar uma estimativa das grandezas variáveis do sistema.

Todas as medições e cálculos baseados em modelos são, de certo modo, estimativas. Sinais ruidosos de sensores, aproximações nas equações que descrevem o comportamento do sistema e fatores externos não considerados introduzem incerteza sobre os valores inferidos para o estado de um sistema. O filtro de Kalman combina uma predição do estado de um sistema com uma nova medida usando uma média ponderada. A ideia dos pesos é que valores com menor incerteza estimada sejam mais "confiados". Os pesos são calculados através da covariância, uma medida da incerteza estimada da predição do estado do sistema. O resultado da média ponderada é uma nova estimativa do estado, que se localiza entre o estado predito e o estado medido, apresentando uma melhor incerteza estimada que qualquer um dos dois unicamente. Este processo é repetido a cada passo de tempo, com a nova estimativa e sua covariância gerando a predição usada na próxima iteração. Isto significa que o filtro de Kalman funciona recursivamente e requer apenas a última estimativa, e não o histórico completo, do estado de um sistema para calcular o próximo estado.

2.5. Técnica aplicada nesta dissertação

No desenvolvimento de um sistema de navegação de um AUV deve-se ter em conta certos aspetos como precisão, custo e energia necessária ao funcionamento do veículo. De forma a garantir que estas condições sejam respeitadas, optou-se por um sistema sem sincronização de relógio e com uma configuração de OWTT. Para estimar a posição em tempo real o veículo está equipado com um sensor de pressão, que calcula diretamente a profundidade do veículo e um sistema acústico para posicionamento LBL.

Capítulo 3

Algoritmo matemático

Neste capítulo irei, inicialmente, apresentar a arquitetura da solução utilizada nesta dissertação, assim como, a quantidade do número de boias acústicas e de que forma se estimou a posição do AUV.

Após esta análise, é descrito de que forma procedi até obter um conjunto de equações capazes de, com algum erro, calcular a posição do veículo.

No fim deste capítulo, encontram-se as limitações do algoritmo matemático e de que forma as consegui superar.

3.1. Arquitetura da solução

Para um sistema de posicionamento acústico correto é necessário recorrer a, no mínimo, três boias acústicas, se existir sincronismo dos relógios, e a, no mínimo, quatro, se não existir sincronismo. Esta baliza extra serve de apoio ao cálculo da posição do veículo, o que irá ser explicado mais aprofundadamente no próximo subcapítulo. Como apenas queremos a coordenadas relativas à posição no eixo X e no Y, pois a correspondente à cota, eixo Z, obtém-se diretamente do sensor de pressão, utilizou-se apenas três balizas acústicas num sistema sem sincronismo entre o relógio do veículo e os das boias.

3.2. Desenvolvimento do algoritmo

Para o desenvolvimento do algoritmo matemático foi necessário recorrer à ferramenta Matlab, dada a complexidade das equações a trabalhar, desta forma obtive as equações utilizadas no processo de cálculo da posição do AUV, no desenvolvimento desta dissertação. Após esta etapa, testei estas equações, com dados fictícios, de forma a melhorar o algoritmo.

Como iremos ter um sensor de pressão para calcular a profundidade, podemos projetar este sistema em duas dimensões, recorrendo a um eixo cartesiano ortogonal. Atribui determinadas posições às três balizas, como se pode ver na figura 3.1.

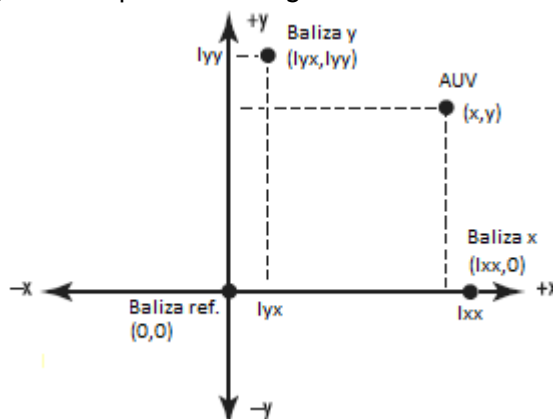


Figura 3.1 - Referencial cartesiano ortogonal com as posições das balizas e do veículo.

Após a implantação das boias acústicas no meio marítimo, e como as suas posições são fixas, é possível traçar este sistema de eixos onde a baliza ref (referência), representada na Figura 3.1, irá ser sempre a origem do referencial, qualquer que este seja, ficando sempre com a posição $(x=0, y=0)$. A baliza x ficará sempre no eixo positivo das abcissas, com uma posição de $(x=l_{xx}, y=0)$, e l_{xx} maior que zero. Por fim temos a baliza y que, poderá tomar qualquer valor real, de forma a obter as equações finais atribui-lhe as coordenadas $(x=l_{yx}, y=l_{yy})$. Este sistema de eixos foi projetado desta maneira de forma a evitar erros no cálculo da posição do veículo, tal como quando a abscissa da baliza x tomava valores menores que zero, a posição do AUV calculada pelo algoritmo era incorreta.

Todos estes valores serão, posteriormente, substituídos pelas coordenadas reais aquando o posicionamento das balizas, o que requer uma calibração extremamente precisa como já se tinha falado anteriormente.

Temos ainda o veículo, representado na Figura 3.1, com uma posição (x, y) , que poderá tomar qualquer valor real.

Através do módulo das distâncias, “ d_r ”, “ d_x ” e “ d_y ”, entre as balizas e o veículo, como se pode ver na Figura 3.2, é possível obter as seguintes equações iniciais representadas nas equações 3.1, 3.2 e 3.3.

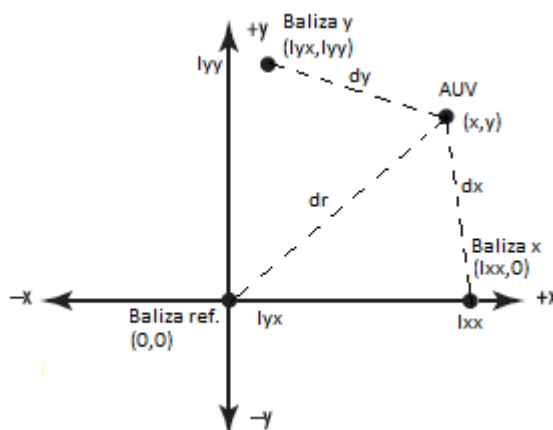


Figura 3.2 - Representação das distâncias entre as balizas e o AUV.

$$|d_r| = \sqrt{x^2 + y^2} \quad (3.1)$$

$$|d_x| = \sqrt{(x - l_{xx})^2 + y^2} \quad (3.2)$$

$$|d_y| = \sqrt{(x - l_{yx})^2 + (y - l_{yy})^2} \quad (3.3)$$

Como não existe sincronismo dos relógios entre o emissor e o recetor, não é conhecido o instante em que é enviado o sinal acústico, apenas quando este foi recebido, representado na equação 3.4.

$$t = t_0 + \frac{d}{v} \quad (3.4)$$

Na **Erro! A origem da referência não foi encontrada.** temos “t” que representa o instante a que foi recebido o sinal pelo veículo, “t₀” o instante em que este sinal foi emitido, “d” a distância da baliza ao veículo e “v” a velocidade de propagação do sinal. Desta forma, para o nosso sistema temos que:

$$t_r = t_0 + \frac{d_r}{v} \quad (3.5)$$

$$t_x = t_0 + \frac{d_x}{v} \quad (3.6)$$

$$t_y = t_0 + \frac{d_y}{v} \quad (3.7)$$

Representado nas equações 3.5, 3.6 e 3.7, temos “t_r”, “t_x” e “t_y” que correspondem ao tempo de receção do sinal da baliza de referência, da baliza x e da baliza y, respetivamente. Estes tempos serão o ponto de partida e os únicos dados fornecidos ao microcontrolador para o cálculo da posição do AUV. Como temos um sistema sem sincronização de relógio, não teremos qualquer informação sobre o instante “t₀”, a que foi emitido o sinal, tornando-se numa incógnita ao sistema, de forma a se conseguir contornar este problema é utilizado uma baliza acústica extra neste tipo de comunicação, onde não existe sincronismo de relógios, referido no subcapítulo **Erro! A origem da referência não foi encontrada..**

Estes dados serão adquiridos aquando a deteção do sinal pelo transdutor do veículo, que é responsável por identificar a que boias acústicas pertence o sinal recebido, este processo é realizado através de um banco de filtros presente na sua constituição que detetam a frequência da onda acústica que é única para cada baliza acústica.

Após a receção do sinal acústico é necessária proceder ao cálculo da posição do veículo, para tal foi necessário calcular expressões matemáticas que o pudessem fazer. Estas equações foram obtidas através da diferença dos instantes “t_r”, “t_x” e “t_y”, representado nas equações 3.8 e 3.9.

$$t_r - t_x = \frac{d_r}{v} - \frac{d_x}{v} \quad (3.8)$$

$$t_r - t_y = \frac{d_r}{v} - \frac{d_y}{v} \quad (3.9)$$

Como se pode verificar nas equações 3.8 e 3.9, o instante “ t_0 ” desaparece e portanto é irrelevante na solução, desta forma é possível calcular a posição do veículo mesmo não sabendo quando foi emitido o sinal.

Substituindo “ d_r ”, “ d_x ” e “ d_y ”, obtém-se:

$$v * (t_r - t_x) = \sqrt{x^2 + y^2} - \sqrt{(x - l_{xx})^2 + y^2} \quad (3.10)$$

$$v * (t_r - t_y) = \sqrt{x^2 + y^2} - \sqrt{(x - l_{yx})^2 + (y - l_{yy})^2} \quad (3.11)$$

De forma a simplificar os cálculos substitui-se “ $v(t_r - t_x)$ ” por “ d_{rx} ” e “ $v(t_r - t_y)$ ” por “ d_{ry} ”, ficando com as equações 3.12 e 3.13.

$$d_{rx} = \sqrt{x^2 + y^2} - \sqrt{(x - l_{xx})^2 + y^2} \quad (3.12)$$

$$d_{ry} = \sqrt{x^2 + y^2} - \sqrt{(x - l_{yx})^2 + (y - l_{yy})^2} \quad (3.13)$$

Manipulando estas expressões obteve-se um conjunto de equações que permitem calcular a posição do veículo, através dos instantes a que são recebidos, pelo veículo, os sinais emitidos pelas balizas. O par de equações usado para calcular a posição do veículo (x,y) encontra-se representado nas equações 3.12 e 3.13.

$$\begin{cases} x_1 = \frac{1}{2} * \left(l_{xx} - d_{rx} \sqrt{\frac{d_{rx}^2 - l_{xx}^2 - 4 * y_1^2}{d_{rx}^2 - l_{xx}^2}} \right) \\ y_1 = \frac{1}{2 * l_{yy}} \left(\alpha + \frac{(d_{ry} * l_{xx} - d_{rx} * l_{yx})(l_{yy} * \kappa + \tau)}{((d_{rx} * l_{xx} - d_{ry} * l_{yy})^2 + l_{yy}^2(d_{rx}^2 - l_{xx}^2))} \right) \end{cases} \quad (3.14)$$

$$\begin{cases} x_2 = \frac{1}{2} * \left(l_{xx} - d_{rx} \sqrt{\frac{d_{rx}^2 - l_{xx}^2 - 4 * y_2^2}{d_{rx}^2 - l_{xx}^2}} \right) \\ y_2 = \frac{1}{2 * l_{yy}} \left(\alpha - \frac{(d_{ry} * l_{xx} - d_{rx} * l_{yx})(l_{yy} * \kappa - \tau)}{((d_{rx} * l_{xx} - d_{ry} * l_{yy})^2 + l_{yy}^2(d_{rx}^2 - l_{xx}^2))} \right) \end{cases} \quad (3.15)$$

Onde “ α ”, “ κ ” e “ τ ” representam, respetivamente:

$$\alpha = d_{ry}(d_{rx} - d_{ry}) + l_{yx}(l_{yx} - l_{xx}) + l_{yy}^2 \quad (3.16)$$

$$\kappa = \sqrt{-(d_{rx} + l_{xx})(d_{rx} - l_{xx})(-d_{ry}^2 + l_{yx}^2 + l_{yy}^2)(-d_{rx}^2 - d_{ry}^2 + 2 * d_{rx} * d_{ry} + l_{xx}^2 + l_{yx}^2 - 2 * l_{xx} * l_{yx} + l_{yy}^2)} \quad (3.17)$$

$$\begin{aligned} \tau = d_{rx} * d_{ry} & \left(d_{rx} * l_{yx} - d_{ry}(l_{xx} + l_{yx}) \right) + l_{xx} * l_{yx} \left(l_{xx} * d_{ry} - l_{yx}(d_{rx} + d_{ry}) \right) \\ & + l_{yy}^2(d_{rx} * l_{yx} - d_{ry} * l_{xx}) + d_{rx} * l_{yx}^3 + d_{ry}^3 * l_{xx} \end{aligned} \quad (3.18)$$

Como se verifica nas equações, existem dois conjuntos de expressões (x_1 , y_1) e (x_2 , y_2) para uma determinada posição do veículo. No subcapítulo 3.3 explicarei como é escolhida a opção correta.

3.3. Limitações e suas soluções

Como visto no subcapítulo 3.2 existem dois conjuntos com duas soluções possíveis para uma determinada posição. A solução obtida através do conjunto de equações 3.12, dá-nos a solução correta para quando o veículo está a operar no primeiro, segundo e quarto quadrante, para qualquer sistema de eixos ortogonal utilizado. Com o conjunto de equações 3.13 é possível obter a solução para quando o AUV está a navegar no terceiro quadrante. Na figura 3.3 encontra-se representado um sistema de eixos ortogonal, onde se podem verificar os quadrantes do mesmo.

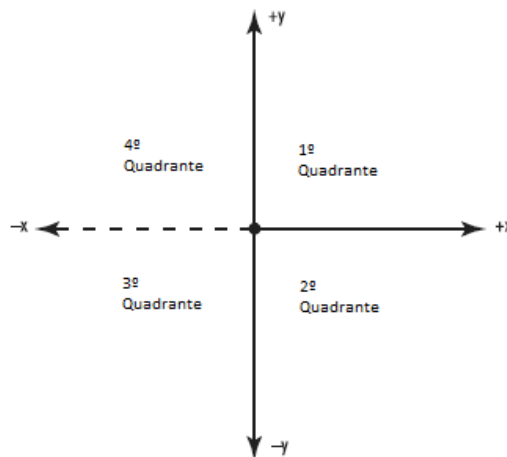


Figura 3.3 - Ilustração dos quatro quadrantes em um referencial cartesiano.

Quando o veículo toma valores de “x” negativos ou zero e “y” igual a zero, simultaneamente, na figura 3.3 esta região de valores encontra-se a tracejado, o algoritmo não consegue calcular a sua posição. Isto deve-se ao facto de que “ d_{rx} ” e “ l_{xx} ” ficarem com o mesmo valor algébrico. Nas equações 3.19 e 3.20 é feita uma demonstração deste problema.

Usando 3.12, com y igual a zero, temos:

$$d_{rx} = \sqrt{x^2 + 0^2} - \sqrt{(x - l_{xx})^2 + 0^2} = |-x| - |(-x) - l_{xx}| = x - x + l_{xx} = l_{xx} \quad (3.19)$$

Que, quando substituído na equação 3.14, necessária para estimar a abcissa do veículo resulta na expressão 3.20, para $d_{rx} = l_{xx}$ e “y” igual a zero.

$$x = \frac{1}{2} * \left(l_{xx} - l_{xx} \sqrt{\frac{l_{xx}^2 - l_{xx}^2 - 4 * 0}{l_{xx}^2 - l_{xx}^2}} \right) = \frac{1}{2} * \left(l_{xx} - l_{xx} \sqrt{\frac{0}{0}} \right) = \text{indeterminado} \quad (3.20)$$

Tendo em conta a expressão 3.20, é possível verificar que para qualquer valor de “x” negativo ou zero com “y” igual a zero, torna-se impossível estimar o seu valor pois o seu valor fica indeterminado. Desta forma o algoritmo não consegue calcular valores desta natureza, no entanto quando “y” toma valores muito ligeiramente maiores que zero, na ordem dos micrómetros, o sistema já consegue estimar a posição do veículo.

De maneira a resolver este problema, foi calculado um novo conjunto de equações para “x” e “y”, numerado de 3.21, utilizado especificamente quando o AUV toma os valores representados pela linha a tracejado da figura 3.3.

$$\begin{cases} x = \left(\frac{d_{rx} (d_{ry} (-l_{xx} (l_{yx}^2 + l_{xx} * l_{yx} + l_{yy}^2)) + l_{yy} * B) + C}{2 (l_{yy}^2 (d_{rx}^2 - l_{xx}^2) + (d_{rx} * l_{yx} - d_{ry} * l_{xx})^2)} \right) \\ y = \left(\frac{l_{yy} (l_{yx} (2 * x - l_{yx}) + d_{ry}^2 - l_{yy}^2) + d_{ry} * A}{2 * (d_{ry}^2 - l_{yy}^2)} \right) \end{cases} \quad (3.21)$$

Onde “A”, “B” e “C” representam:

$$A = \sqrt{((-d_{ry}^2 + l_{yx}^2 + l_{yy}^2) * (-d_{ry}^2 + l_{yx}^2 - 4 * l_{yx} * x + l_{yy}^2 + 4 * x^2))} \quad (3.22)$$

$$B = \sqrt{\left((l_{xx}^2 - d_{rx}^2) * (-d_{ry}^2 + l_{yx}^2 + l_{yy}^2) * \left(d_{rx} * d_{ry} * \left(2 - \frac{d_{rx}}{d_{ry}} - \frac{d_{ry}}{d_{rx}} \right) + l_{xx} * (l_{xx} - 2 * l_{yx}) + l_{yx}^2 + l_{yy}^2 \right) \right)} \quad (3.23)$$

$$C = d_{rx}^2 * (l_{xx} * l_{yy}^2 + l_{yx} * (l_{yx}^2 + l_{yy}^2 + d_{rx} * d_{ry})) + d_{ry}^2 * \left(d_{rx}^2 * \left(\frac{d_{ry}}{d_{rx}} * l_{xx} - l_{yx} - l_{xx} \right) + l_{xx}^3 \right) \quad (3.24)$$

Através do conjunto 3.21, é possível calcular a região a tracejado da figura 3.3. Estas expressões foram calculadas utilizando o método inverso das anteriores, ou seja, nos conjuntos 3.14 e 3.15, temos o “x” em função de “y” e neste novo conjunto, 3.21, temos o “y” em função de “x”, desta forma o cálculo da abcissa para os valores em causa, em nada é afetado para qualquer que seja o “y”.

Apesar de ser possível calcular a região a tracejado da figura 3.3, ainda reside um problema, que é a dualidade de soluções, de forma a resolver este problema provisoriamente decidiu-se optar por um sistema que apenas atuará num dos conjuntos dos quadrantes de cada vez, sendo apenas possível alterá-lo manualmente. Este problema foi resolvido através da implementação de um filtro de Kalman, que possibilita uma estimação da posição do veículo recorrendo a outras técnicas que não dependem inteiramente do conjunto de expressões 3.15 e 3.15, e será novamente introduzido num capítulo posterior a este.

Capítulo 4

Software desenvolvido

O objetivo deste capítulo é dar a entender o algoritmo desenvolvido em linguagem C, utilizado para estabelecer comunicação entre a veículo e os *beacons*, e ainda, processar os dados que este recebe de cada boia acústica.

Quando à estrutura do capítulo, em primeiro lugar, será apresentada a solução utilizada no desenvolvimento desta dissertação. De seguida, irá ser demonstrado o funcionamento do algoritmo desenvolvido, assim como as suas propriedades. Por último será demonstrado um conjunto de testes ao algoritmo elaborado.

4.1. Arquitetura utilizada

Aquando da elaboração do algoritmo foi necessário escolher entre uma comunicação através do protocolo TCP ou UDP. Como o sistema a ser projetado requer que o servidor, também conhecido por AUV, escute todos os dados enviados pelos clientes, neste caso, as boias acústicas, optou-se por uma arquitetura usando o protocolo UDP, com *Datagram Sockets*, para estabelecer comunicação. Desta forma, como não existe garantia de que os dados foram entregues corretamente, é necessário implementar rotinas no algoritmo necessárias para verificar a existência de erros, e os solucionar da melhor maneira possível, as quais irão ser abordadas no subcapítulo 4.2.

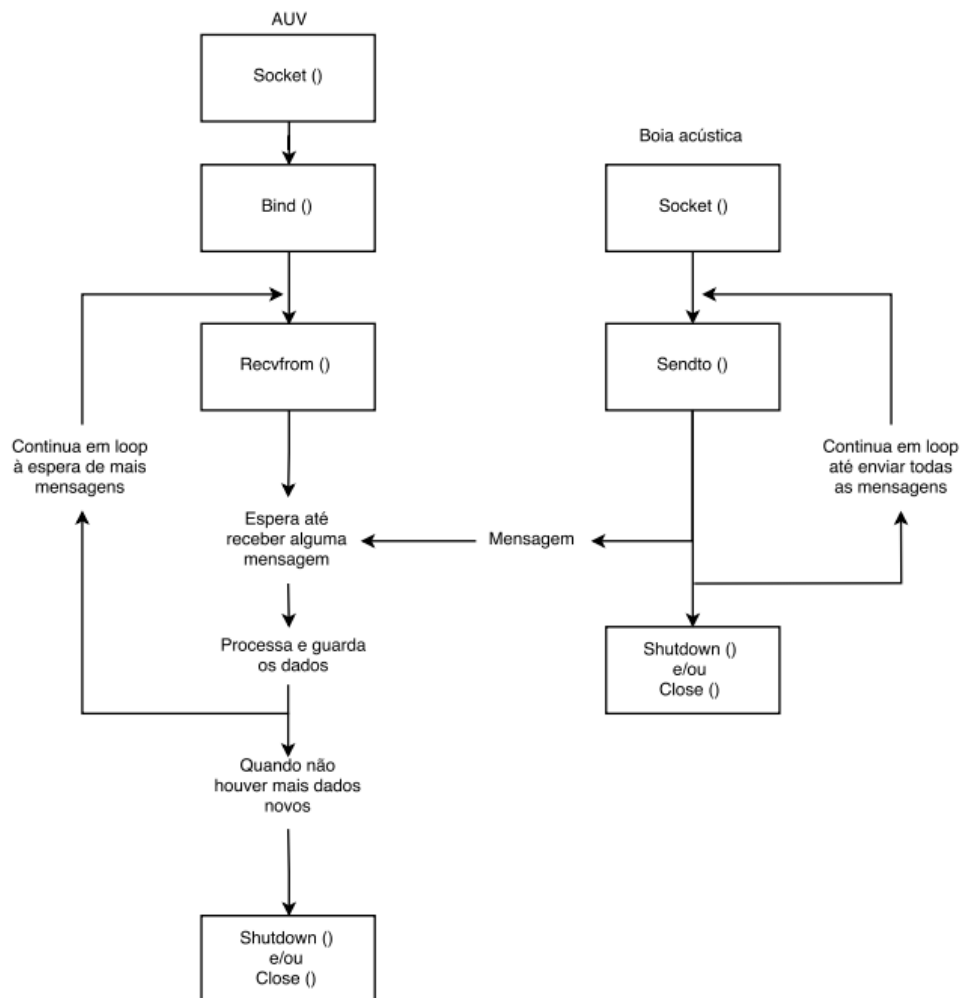


Figura 4.1 - Diagrama da comunicação, entre o AUV e uma boia acústica, implementada.

Como é possível verificar na figura 4.1, a forma de como o sistema desenvolvido comunica é um pouco diferente de uma configuração típica, representada pela figura 2.15. Esta diferença reside no facto de não existir uma resposta por parte do servidor/veículo, de uma mensagem de *acknowledge*, no momento em que recebe um sinal proveniente de um cliente/boia acústica. Esta propriedade leva a que, como visto em [4], a duração da bateria do AUV se torne significativamente maior, a uma diminuição de sinais trocados, o que diminui a quantidade de erros provocados pela perda de sinais e um ambiente mais “limpo”.

O veículo escuta a(s) boia(s), até que não haja mais sinais novos num dado intervalo de tempo significativo ou quando existe uma ordem de paragem. Os dados recebidos e os obtidos através das expressões mencionadas no capítulo 3 são guardados e extraídos para documentos de texto, que posteriormente serão analisados.

4.2. Mecanismos para evitar erros

Em caso de exemplo, vou supor que o veículo já recebeu os seguintes dados, presentes na tabela 4.1, onde SPR significa, sinal por receber.

	<i>Beacon 1</i>	<i>Beacon 2</i>	<i>Beacon 3</i>
Instante (segundos)	0,010	0,015	0,020
	1,015	1,022	SPR

Tabela 4.1 - Dados recebidos pelo veículo.

Como seria de esperar, se não existissem perdas de sinais, o próximo sinal acústico detetado, pelo veículo, teria origem no *beacon 3*. No entanto, vou considerar que o próximo sinal é detetado no instante 2,020 segundos e tem origem no *beacon 1*. Tendo isto em conta pode-se considerar que o sinal do *beacon 3* foi perdido, como representado na tabela 4.2.

Como o veículo se desloca com uma velocidade máxima de 2 ms^{-1} , tem-se uma variância máxima de 2 metros em relação à posição anterior, desta forma, assume-se que, quando existe perda de um ou mais sinais, a posição atual é igual à posição anterior, como representado na tabela 4.3, pois não existem dados suficientes para a calcular se existir perda de pelo menos um dos sinais acústicos, considera-se, também, o(s) instante(s) perdido(s) como zero, como representado na tabela 4.2.

	<i>Beacon 1</i>	<i>Beacon 2</i>	<i>Beacon 3</i>
Instante (segundos)	0,010	0,015	0,020
	1,015	1,022	0
	2,020	SPR	SPR

Tabela 4.2 - Dados recebidos pelo veículo com perda do *beacon 3*.

	Abcissa (metros)	Ordenada (metros)
Conjunto 1	17,75	40,91
Conjunto 2	17,75	40,91

Tabela 4.3 - Posição do veículo, com perda do *beacon 3* do segundo conjunto de dados.

Outra situação seria, supondo que teríamos recebido os dados presentes na tabela 4.4.

	<i>Beacon 1</i>	<i>Beacon 2</i>	<i>Beacon 3</i>
Instante (segundos)	0,010	0,015	0,020
	1,020	1,022	SPR

Tabela 4.4 - Dados recebidos pelo veículo.

O próximo sinal que deveríamos receber seria do *beacon 3*, no instante 1,030 segundos, no entanto, esse sinal foi perdido e foi escutado o sinal do próximo conjunto, para o mesmo *beacon* no instante 2,040 segundos. Tal como no caso anterior, este último sinal recebido é guardado, enquanto que o sinal perdido fica a zero, como na tabela 4.5.

	<i>Beacon 1</i>	<i>Beacon 2</i>	<i>Beacon 3</i>
Instante (segundos)	0,010	0,015	0,020
	1,015	1,022	0
	SPR	SPR	2,040

Tabela 4.5 - Dados recebidos pelo veículo com perda do *beacon 3*, do segundo conjunto.

Existe ainda outra situação que deve ser tratada de uma forma diferente, que é quando existe perda de um dos sinais no primeiro conjunto de dados. Supondo que o veículo já escutou

os dados presentes na tabela 4.6. Esta situação tem uma resolução diferente das anteriores pois como é a primeira vez que o veículo está a calcular a posição, não sabe onde se encontra, logo não é possível comparar com nenhuma anterior.

	<i>Beacon 1</i>	<i>Beacon 2</i>	<i>Beacon 3</i>
Instante (segundos)	SPR	0,015	0,020

Tabela 4.6 - Dados recebidos, do primeiro conjunto, pelo veículo.

Supondo que o próximo valor irá pertencer ao *beacon 2*, com o instante 1,022 segundos, o algoritmo, é capaz de verificar que existiu perda do sinal do *beacon 1*, o mesmo aconteceria se recebesse um sinal do *beacon 1* ou 3. Desta forma, desprezasse todos os valores recebidos e guarda-se o último recebido, como representado na tabela 4.7.

	<i>Beacon 1</i>	<i>Beacon 2</i>	<i>Beacon 3</i>
Instante (segundos)	SPR	1,022	SPR

Tabela 4.7 - Dados recebidos pelo veículo, com perda no conjunto inicial.

Através de um vasto conjunto de condições, como as que estão representadas acima, é possível evitar todo o tipo de erros deste género, tornando o algoritmo extremamente robusto e fiável.

4.3. Testes do algoritmo

Para uma velocidade em x e y de 1 ms^{-1} , e um conjunto de 100 instantes, de cada *beacon*, escutados por parte do AUV. Assumindo que l_{xx} toma o valor de 50 m, l_{yx} de 10 m e l_{yy} de 100 m, tendo em conta a figura 4.2.

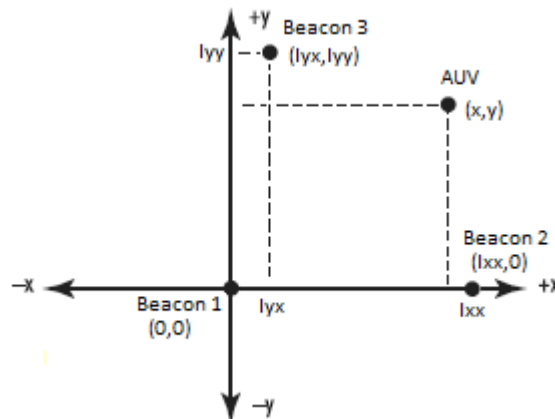


Figura 4.2 - Referencial usado nos testes.

Com isto, é possível calcular uma série de posições utilizando o algoritmo matemático descrito no capítulo 3. Com a posição inicial de (10,10) (m), o veículo realizou uma trajetória como representado na figura 4.3.

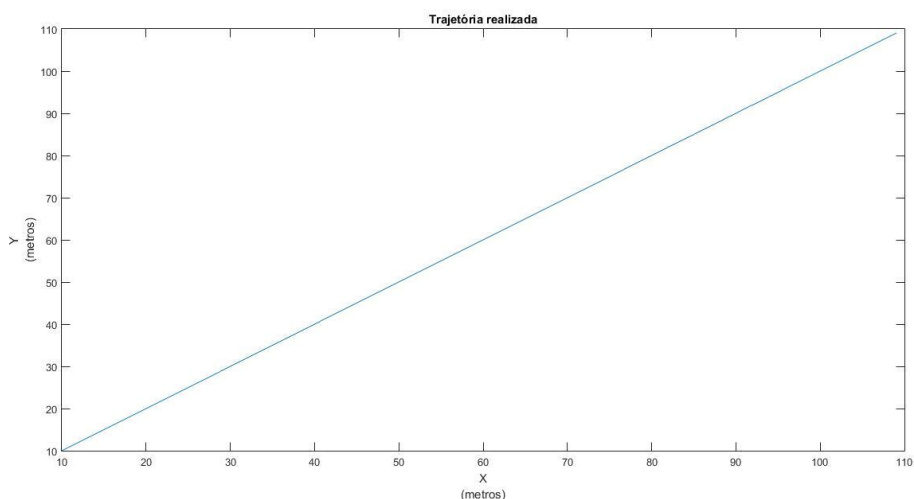


Figura 4.3 - Trajetória do veículo.

Tendo em conta que irei simular as *beacons* com um software concebido para enviar os instantes de tempo a que os sinais iriam ser recebidos pelo veículo e a que boia pertencem, tal como o AUV os iria receber. Como a comunicação através de UDP acarreta alguns erros poderei não receber exatamente os mesmos intervalos de tempo que enviem através do cliente, para isso, foi inserido nos testes um conjunto de gráficos que correspondes a essa diferença, como representado na figura 4.4.

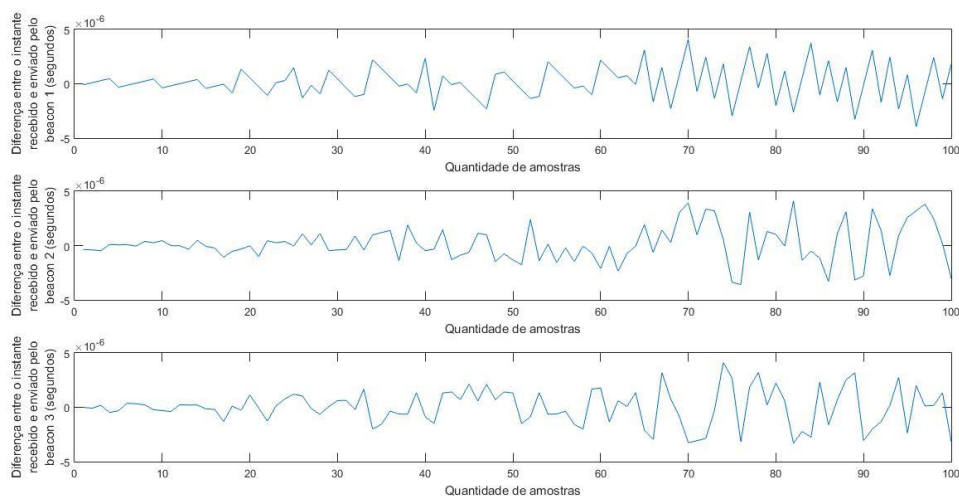


Figura 4.4 - Diferença entre os instantes emitidos e recebidos.

Através da figura 4.4, verifica-se que existe uma diferença que atinge no máximo valores na ordem dos 5 μs , que com uma velocidade do som, na água, de 1500 ms^{-1} corresponde a uma distância de 7,5 mm, que é desprezável.

Na figura 4.5 está representado a diferença entre a posição calculada com o algoritmo e a posição real.

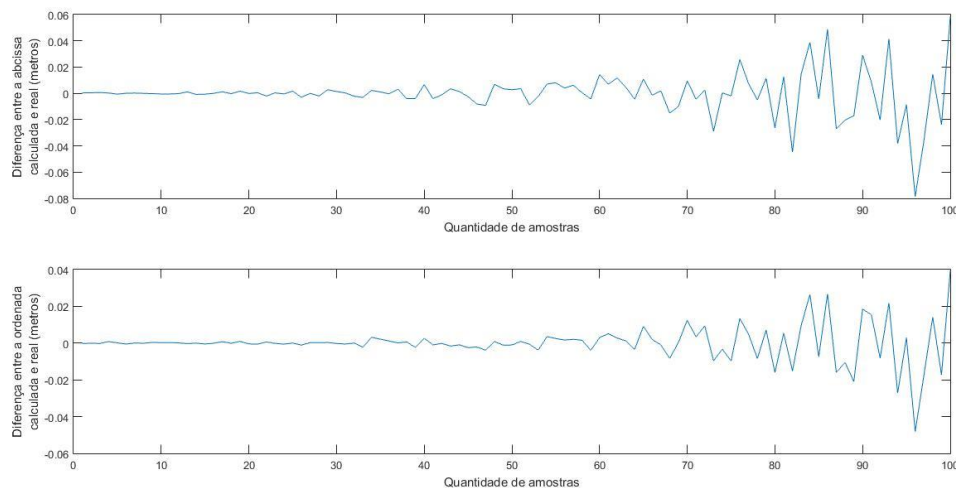


Figura 4.5 - Diferença entre a posição calculada e a real.

Examinado a figura 4.5, verifica-se que o erro absoluto máximo da abcissa é de 80 mm, e da ordenada de 40 mm, que como no caso anterior é desprezável.

Efetuuou-se mais um teste de forma a descobrir a partir de que distância se começa a ter erros elevados, ou seja, para perceber qual a melhor área de ação do algoritmo, também conhecido como *Baseline*. Para isso, neste teste foi utilizado o mesmo referencial, com $v_x = -2 \text{ ms}^{-1}$ e $v_y = 1 \text{ ms}^{-1}$, com posição inicial em $x_0 = 10 \text{ m}$ e $y_0 = 10 \text{ m}$, para um conjunto de 200 posições calculadas, ou seja, o x irá atingir o valor máximo de -390 m e o y de 210 m.

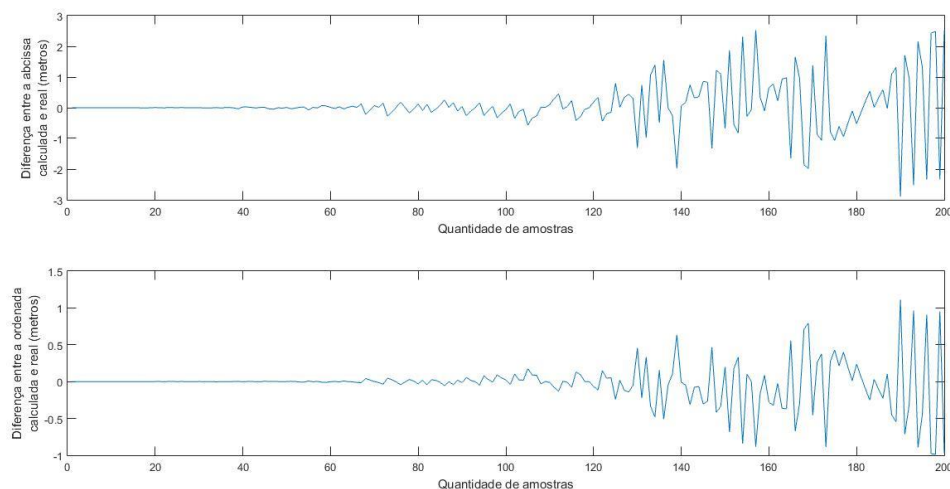


Figura 4.6 - Diferença entre a posição calculada e a real.

Tendo em conta os resultados obtidos na figura 4.6, repara-se que a partir da amostra 140, que corresponde a uma abcissa na ordem dos -270 m e uma ordenada de 150 m, os erros chegam até 3 m, no entanto para a ordenada o erro máximo foi de 1 m. De forma a entender melhor estes resultados realizou-se um segundo teste, onde se manteve tudo, à exceção da velocidade, que ficou $v_x = 1 \text{ ms}^{-1}$ e $v_y = -2 \text{ ms}^{-1}$.

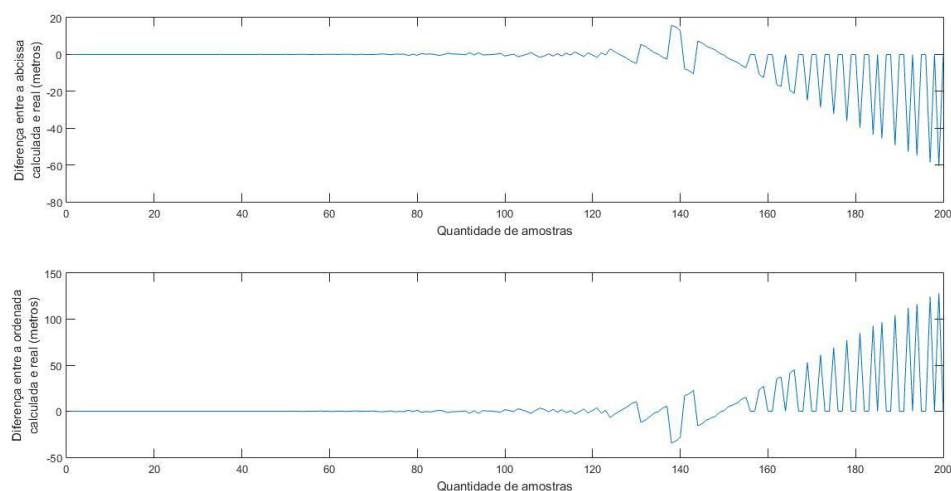


Figura 4.7 - Diferença entre a posição calculada e a real.

Analisando os resultados deste segundo teste, representados na figura 4.7, verifica-se que, novamente, a partir da amostra 140, o que corresponde a uma ordenada na ordem dos -270 m e uma abscissa de 150 m, os erros são inaceitáveis, e como a abscissa depende da ordenada para ser calculada, como se vê nas expressões 3.14 e 3.15, estes erros propagam-se para a abscissa, contaminando o resultado final com um erro totalmente inaceitável.

Realizou-se um terceiro ensaio com tudo igual à exceção da posição dos *beacons*, ficando com l_{xx} igual a 100 m, l_{yx} igual a 0 m e l_{yy} igual a 10 m. Desta vez com o intuito de perceber o quanto o erro diminuía se se aproximasse ligeiramente os *beacons* da posição final de “x” e “y”. Os resultados estão expressos na figura 4.8.

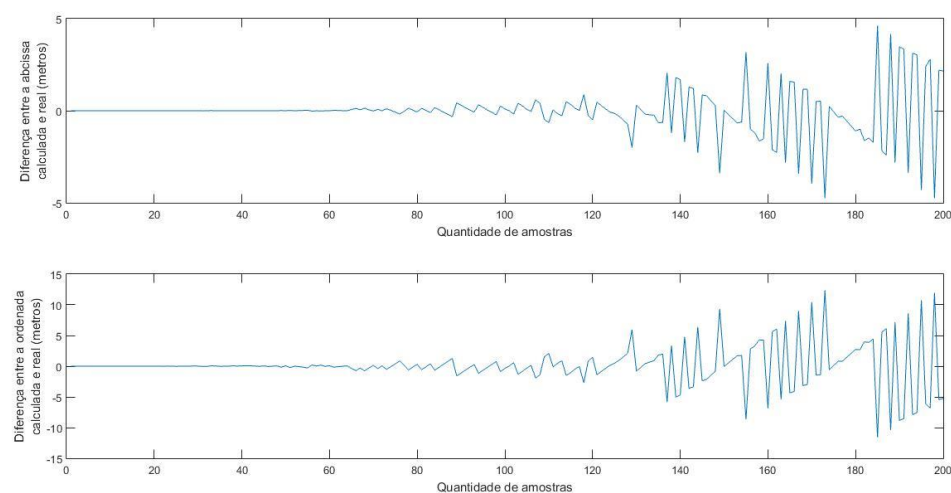


Figura 4.8 - Diferença entre a posição calculada e a real.

Observando a figura 4.8 repara-se que para esta posição dos *beacons* o erro diminui drasticamente, ainda é extremamente elevado, mas em comparação é um resultado muito positivo.

Com estes três testes concluiu-se que a baseline do método utilizado, ou seja, para que valores o algoritmo calcula a posição com poucos erros, seria a parte interior da zona a

tracejado representada na figura 4.9, e para valores fora deste intervalo os erros associados às suas medições de posição irão ser extremamente altos.

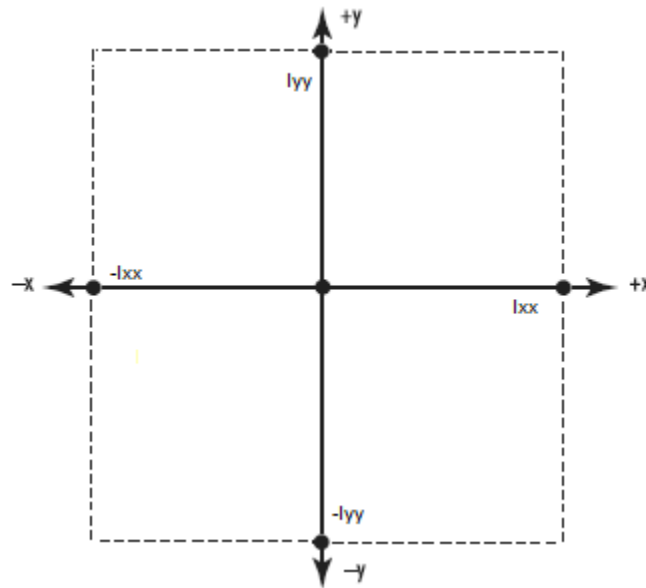


Figura 4.9 - Baseline do algoritmo utilizado.

De notar que este conjunto de resultados não apresentam qualquer tipo de perda de dados, no capítulo 6 será demonstrado um vasto conjunto de testes, onde já haverá falhas no envio de dados, maiores erros no calculo de posição e diferentes trajetórias de forma a demonstrar a robustez do algoritmo desenvolvido.

Capítulo 5

Filtro de Kalman

Este capítulo tem como objetivo, a demonstração da aplicação do filtro de Kalman, elaborado neste trabalho. Inicialmente, irei introduzir um pouco este tema, de uma forma sucinta e clara, explicando as características deste tipo de filtro, e que vantagens traz a este tipo de sistema. De seguida, será apresentado de que forma se aplicou o filtro, com a ajuda de alguns exemplos, assim como as quais as matrizes usadas e os passos que se tomou. Por último irão ser apresentados alguns resultados.

5.1. Desenvolvimento do filtro

Ao realizar os cálculos para o filtro, a estimativa do estado e as covariâncias são representadas por matrizes, para tratar as múltiplas dimensões envolvidas num único passo do cálculo. Desta forma, é possível representar as relações lineares entre diferentes variáveis de estado (como posição e velocidade) em qualquer um dos modelos de transição ou covariâncias.

O algoritmo utilizado para estimar a posição num plano horizontal é baseado num filtro de Kalman. Este algoritmo recorre a um simples modelo do movimento do veículo, para prever a evolução da sua posição, tendo apenas em conta a velocidade e o sentido do movimento, e corrigir a estimativa da posição calculada. Desta forma, o algoritmo faz um melhor uso de toda a variedade de informação existente.

De forma a clarificar a explicação do filtro, irei apresentar a forma mais simples do algoritmo de navegação, que é, quando este apenas estima a posição do veículo, como visto em [6].

O estado do filtro $E = [x \ y]$ é composto pela estimativa das coordenadas, norte e este, do veículo. Além deste estado, o algoritmo também guarda uma matriz P , a matriz de covariância do erro, que mede a covariância do erro de cada estimativa.

Entre a receção de duas medidas consecutivas da posição do veículo, a evolução dos estados E e P regem-se pelas seguintes equações:

$$\dot{E} = F(E, \omega, \psi) = \begin{bmatrix} u(\omega, \psi) \\ v(\omega, \psi) \end{bmatrix} \quad (5.1)$$

$$\dot{P} = \frac{\partial F}{\partial E} * P + P * \frac{\partial F^T}{\partial E} + Q \quad (5.2)$$

Nestas equações ω e ψ , representam, respetivamente, a velocidade do veículo e o sentido do movimento do veículo. As coordenadas da posição do AUV são dadas por u e v , que por sua vez são calculadas em função de ω e ψ . Q é uma matriz simétrica semidefinida positiva, e representa a taxa de aumento do erro devido ao facto de que o veículo não se move exatamente como a equação 5.1. Esta matriz é utilizada para controlar a velocidade de resposta do filtro.

Os valores de E e P são atualizados à mesma taxa do principal ciclo de controlo do movimento do veículo, ou seja, a 10 Hz, o que corresponde a um período de 1 s.

Sempre que uma nova posição do veículo é calculada e validade, o estado E e a matriz de covariância P têm de ser atualizados de acordo com as expressões 5.3 e 5.4, respetivamente.

$$E^+ = E^- + K * (r - r^*) \quad (5.3)$$

$$P^+ = P^- - K * H * P^- \quad (5.4)$$

Onde E^+ e P^+ representam os valores de E e P depois da atualização, enquanto que E^- e P^- representam esses mesmo valores antes da atualização. As coordenadas (x, y) calculadas são representadas por r enquanto que r^* representa os valores reais de (x, y) para cada medição.

As matrizes H e K são dadas pelas equações 5.5 e 5.6, respetivamente.

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.5)$$

$$K = P^- * H^T * S^{-1} \quad (5.6)$$

Onde S é obtido através da expressão 5.7.

$$S = H * P^- * H^T + R \quad (5.7)$$

5.2. Testes ao filtro

Tendo em conta que, para obter uma correta estimação do filtro de Kalman é necessário que se inicializar corretamente E e P , procedeu-se da seguinte forma, expressa pelas equações 5.8 e 5.9, onde x_0 e y_0 corresponde à posição inicial do veículo, que será obtida através das equações mencionadas no capítulo 3, para o primeiro conjunto, completo, de valores escutados pelo veículo.

$$E = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (5.8)$$

$$P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5.9)$$

Após a inicialização do estado E e da matriz de covariância P é necessário “dar” valores corretos para as matrizes Q e R , representadas em 5.10 e 5.11.

$$Q = \begin{bmatrix} q_{vx} & 0 \\ 0 & q_{vy} \end{bmatrix} \quad (5.10)$$

$$R = \begin{bmatrix} r_{xx} & 0 \\ 0 & r_{yy} \end{bmatrix} \quad (5.11)$$

Q representa a confiança que se tem nos valores descritos pela equação de movimento 5.1, ou seja, representa o quão correta a velocidade do som na água utilizada no cálculo é. Como visto em [1], esta velocidade é algo que é possível estimar com uma elevada precisão, logo optou-se por uma matriz como a expressa na equação 5.12.

$$Q = \begin{bmatrix} 0,1 & 0 \\ 0 & 0,1 \end{bmatrix} \quad (5.12)$$

R representa a confiança nos valores calculados pelas expressões mencionadas no capítulo 3 (3.14 e 3.15). Como visto em [1] e nos resultados obtidos no subcapítulo 4.4, os erros obtidos através de sistemas de posicionamento, são quase todos menores que 1 m, portanto optou-se por uma matriz como a representada em 5.13.

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.13)$$

Para a trajetória descrita pela figura 5.1, a mesma do subcapítulo 4.4, assim como a posição inicial, o referencial utilizado e a posição dos *beacons*, o filtro de Kalman obteve os resultados expressos na figura 5.2.

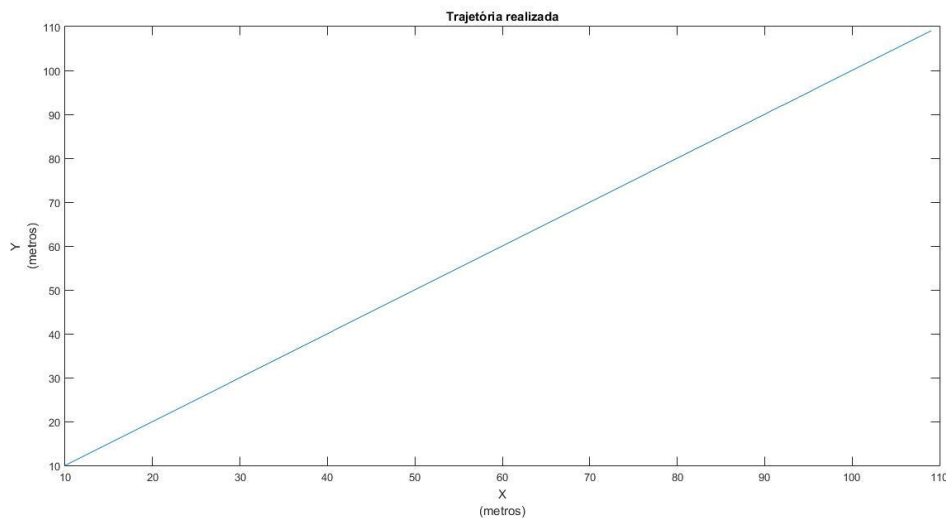


Figura 5.1 - Trajetória realizada.

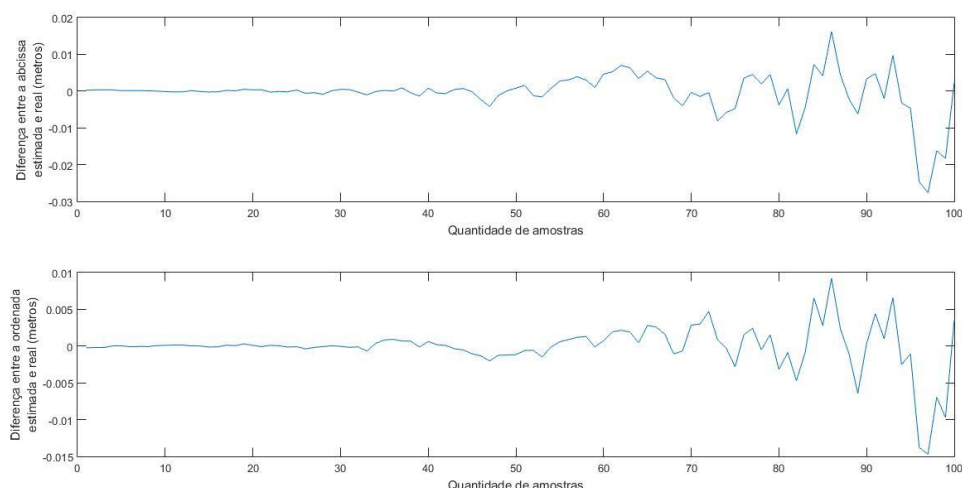


Figura 5.2 - Diferença entre a posição estimada pelo filtro de Kalman e a real.

Através da figura 5.2 verifica-se que o maior erro absoluto da abcissa é de 30 mm e da ordenada é de 15 mm, enquanto que o maior erro absoluto da abcissa e da ordenada, recorrendo apenas às equações 3.14 e 3.15, são de 80 mm e 40 mm, respetivamente, como visto no subcapítulo 4.4.

De notar que estes valores foram calculados nas mesmas circunstâncias do subcapítulo 4.4, ou seja, com a mesma posição dos *beacons* e o mesmo referencial, como o visto na figura 4.5.

Através destes resultados é possível concluir que o filtro de Kalman utilizado apresenta um resultado positivo, diminuindo em mais de metade os erros associados à estimação da posição do veículo.

Este filtro proporciona outra vantagem, com grande importância, que é na decisão de qual o conjunto de equações 3.14 e 3.15 usar, ou seja, através do estado E é possível verificar se o algoritmo irá mudar quadrante e desta forma conseguir atuar em todos os quatro quadrantes em simultâneo. Na figura 5.3 está representada esta situação. Com uma posição inicial de (10, 10) (m) e v_x e v_y igual a -1 ms^{-1} , a trajetória realizada foi a expressa na figura 5.3.

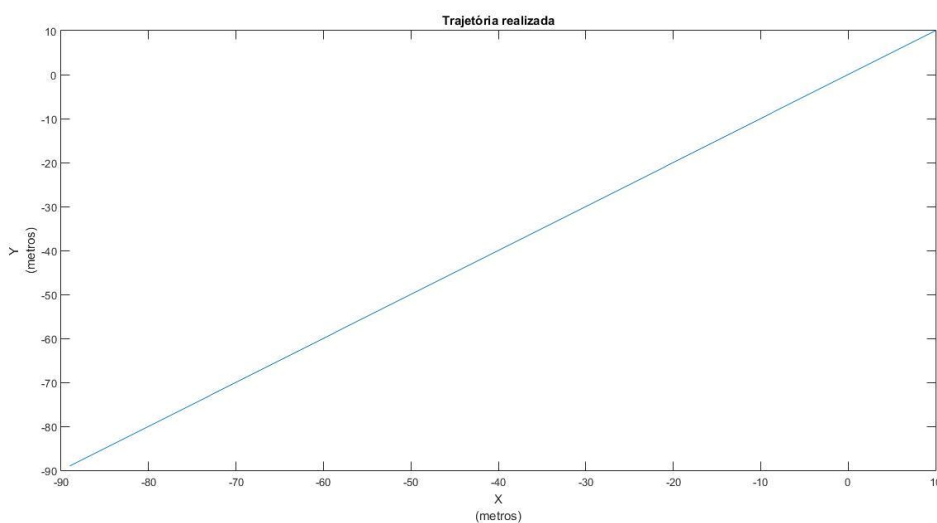


Figura 5.3 - Trajetória realizada com mudança de quadrante.

Na figura 5.4 encontra-se representado a diferença entre os instantes emitidos e os recebidos pelo veículo.

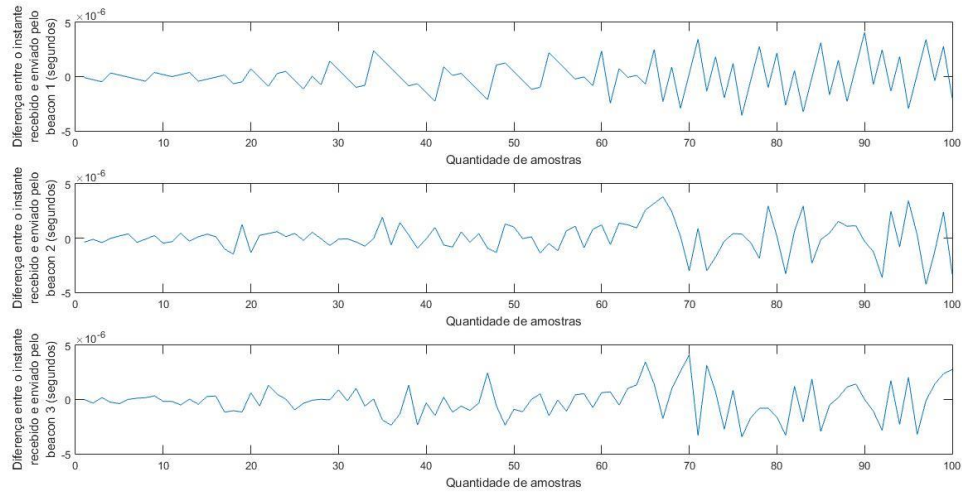


Figura 5.4 - Diferença entre os instantes emitidos e os recebidos.

Verifica-se que os erros relativos aos instantes emitidos e recebidos se mantêm praticamente constantes, tendo em conta os apresentados anteriormente da figura 4.7, logo são desprezáveis.

Na figura 5.5 está representada a diferença entre as coordenadas estimadas pelo filtro e as reais.

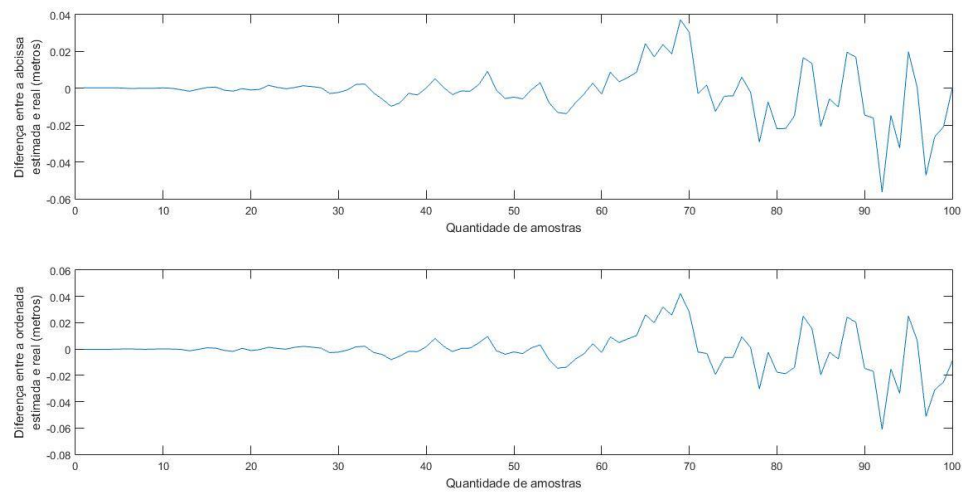


Figura 5.5 - Diferença entre as coordenadas estimadas pelo filtro de Kalman e as reais.

Mais uma vez se verifica que os erros se mantêm praticamente iguais aos apresentados anteriormente na figura 5.2.

Na figura 5.6 está representada diferença entre a posição calculada pelo algoritmo matemático e a real.

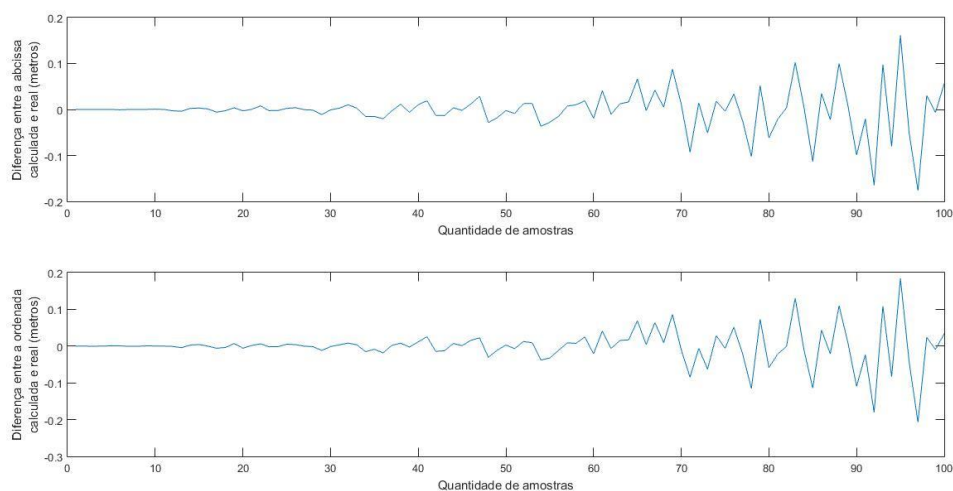


Figura 5.6 - Diferença entre a posição calculada pelo algoritmo matemático e a real.

Desta forma, conclui-se que utilizar o filtro de Kalman para tornar o algoritmo mais inteligente, do ponto de vista de qual expressão matemática a utilizar é benéfico uma vez que não acarreta uma maior quantidade de erros.

De notar que este conjunto de resultados não apresentam qualquer tipo de perda de dados, no capítulo 6 será demonstrado um vasto conjunto de resultados, onde já haverá falhas no envio de dados, maiores erros no calculo de posição e diferentes trajetórias de forma a demonstrar a robustez do algoritmo desenvolvido.

Capítulo 6

Resultados

Neste capítulo irão ser apresentados os resultados do trabalho desenvolvido, de forma a demonstrar a sua robustez e eficiência em ambientes com elevado ruído e falhas. Os testes realizados serão todos diferentes, variando na trajetória realizada, ou seja, na sua velocidade, assim como irão ser introduzidos erros nesses mesmos testes para verificar se o software reage da forma correta, por último serão feitos testes com diferentes posições dos *beacons*. No final será feita uma análise global dos resultados.

Na elaboração destes testes foi sempre tomado em conta que o veículo navegava dentro da área ótima, ou seja, dentro do baseline, onde os erros de cálculo e de estimação da posição do veículo apresenta menos erro, como referido no subcapítulo 4.3 e representado na figura 6.1, quanto maior l_{yy} e l_{xx} maior a área de operação do veículo de forma precisa.

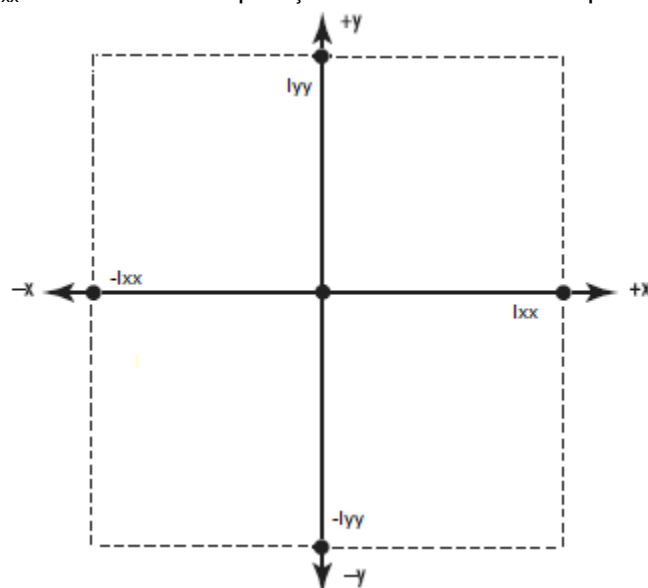


Figura 6.1 - Zona ótima de navegação.

As diferenças entre os instantes enviados e recebidos pelo sistema não foram levadas em conta, pois, o seu valor se manteve praticamente constante em todos os casos de estudo.

Foi ainda levada em conta a velocidade máxima do veículo de 2 ms^{-1} .

6.1. Testes com diferentes posições dos *beacons*

Este grupo de ensaios serve para ilustrar como a posição dos *beacons* influencia a eficácia do algoritmo, pois altera a *baseline*, ou seja, a área ótima de atuação. Irei usar sempre a mesma trajetória que será baseada no referencial da figura 6.2.

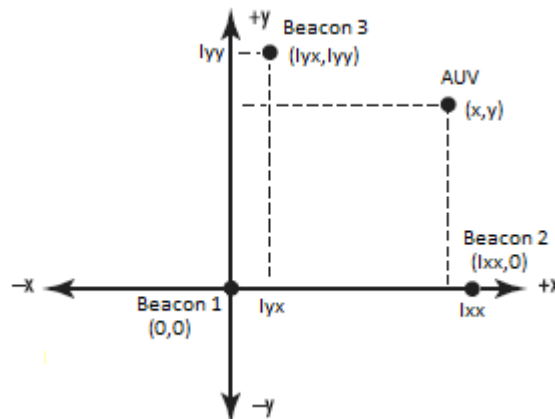


Figura 6.2 - Referencial utilizado nos testes.

O movimento do veículo será feito com $v_x = -1 \text{ ms}^{-1}$ e $v_y = 1,5 \text{ ms}^{-1}$, e posição inicial em $x_0 = 200 \text{ m}$ e $y_0 = -400 \text{ m}$, para um conjunto de 1000 posições calculadas. Utilizou-se esta trajetória para forçar o veículo a passar em três quadrantes distintos, inclusive pelo terceiro quadrante, que implica trocar o conjunto de equações responsável pelo cálculo.

No primeiro exemplo vou assumir que a posição dos *beacons* é tal que $l_{xx} = 500 \text{ m}$, $l_{yx} = 0 \text{ m}$ e $l_{yy} = 500 \text{ m}$.

Na figura 6.3, 6.4 e 6.5 encontra-se a trajetória feita pelo veículo, a diferença entre a posição calculada e a real e a diferença entre posição estimada e a real, respetivamente.

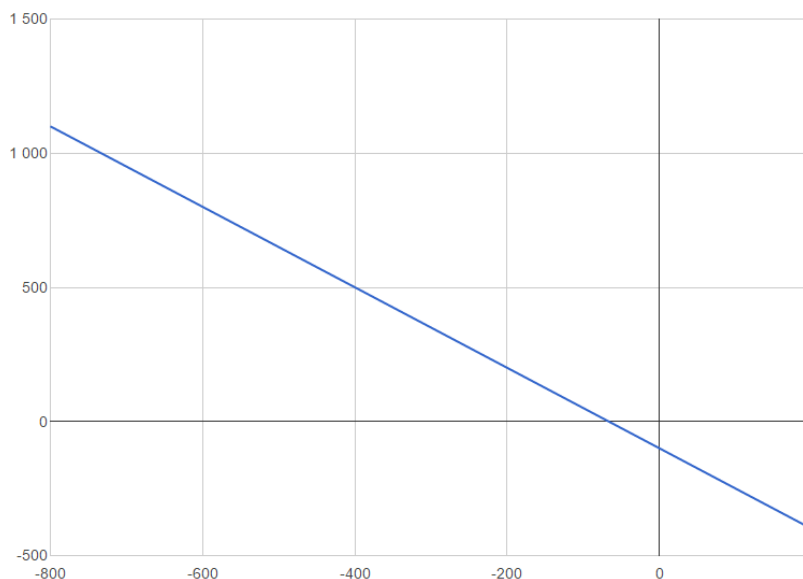


Figura 6.3 - Caminho percorrido pelo veículo para testes de posição das balizas.

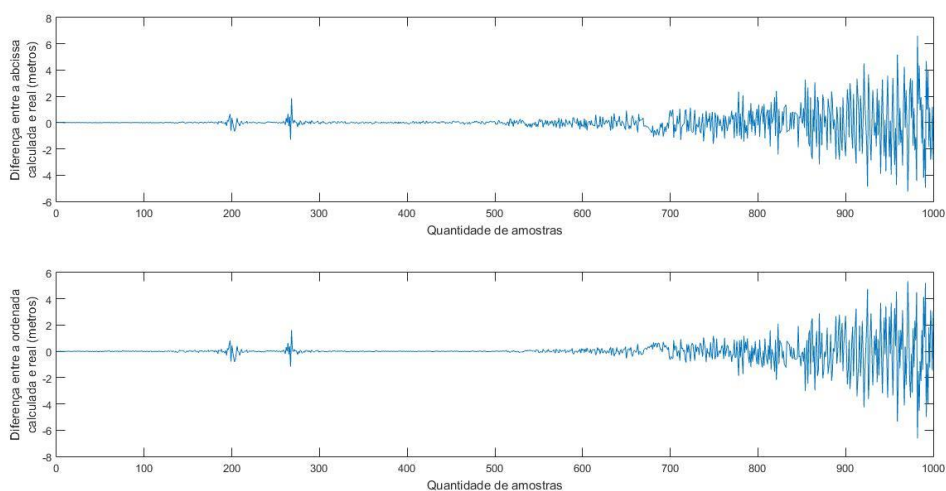


Figura 6.4 - Diferença entre a posição calculada pelo algoritmo e a real.

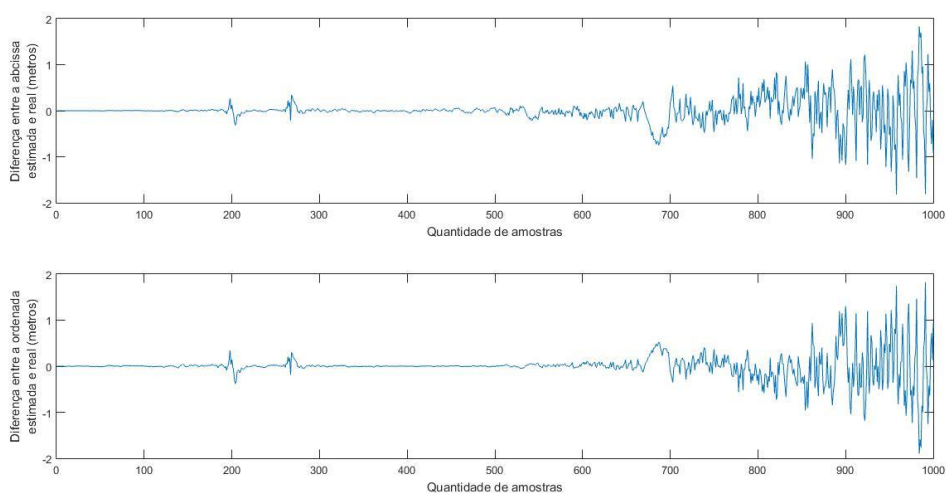


Figura 6.5 - Diferença entre a posição estimada pelo filtro e a real.

Como seria de esperar o ruído começa a ficar elevado a partir da amostra 700, que é quando a abscissa atinge o valor de -500 m e a ordenada de 650 m, desta forma todos os valores seguintes a este, ficam de fora da *baseline*, em questão, para este teste.

No seguinte teste posicionaram as balizas em locais mais distantes da origem que os anteriores, com $l_{xx}=300$ m, $l_{yx}=0$ m e $l_{yy}=300$ m, desta forma todos os valores ficaram dentro da área pretendida. Os seus resultados estão expressos nas figuras 6.6 e 6.7.

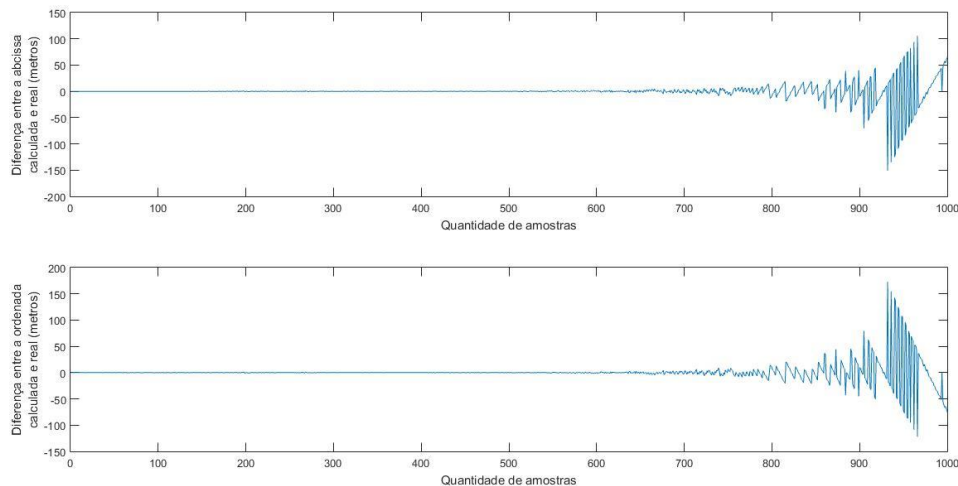


Figura 6.6 - Diferença entre a posição calculada pelo algoritmo e a real.

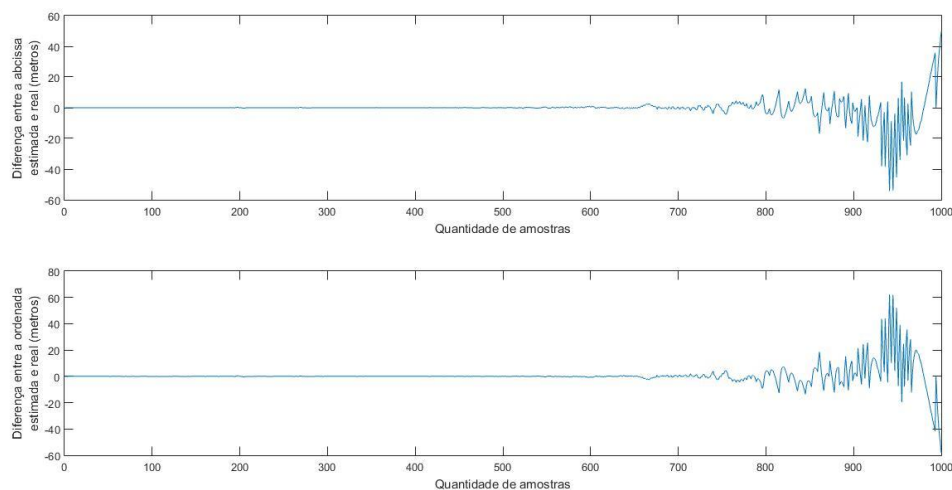


Figura 6.7 - Diferença entre a posição estimada pelo filtro e a real.

Analisando a figura 6.6 e 6.7, é possível verificar que o ruído destes dados é muito elevado para ser desprezável, especialmente a partir da amostra 800.

Através da análise dos testes efetuados neste subcapítulo e, da comparação do resultado dos dois testes, conclui-se que, a posição dos *beacons* afeta, em muito, a eficácia do sistema desenvolvido, como visto anteriormente no subcapítulo 4.3. Este fenómeno é causado quando o veículo começa a navegar fora de uma área ótima definida por l_{xx} e l_{yy} , representada na figura 6.1.

6.2. Testes com diferentes velocidades

Neste conjunto de testes o fator predominante a ser observado é a reação do sistema face à alteração da velocidade de movimento do veículo. Para isso foi utilizado sempre a mesma posição das balizas, l_{xx} igual a 500 m, l_{yx} igual a 0 m e l_{yy} igual a 500 m, e o referencial representado pela figura 6.2, com um conjunto de 1000 amostras.

Desta análise pretende-se retirar informações sobre qual a melhor velocidade para navegar o veículo, desta forma, manteve-se a mesma trajetória, representada na figura 6.8, nos casos estudados.

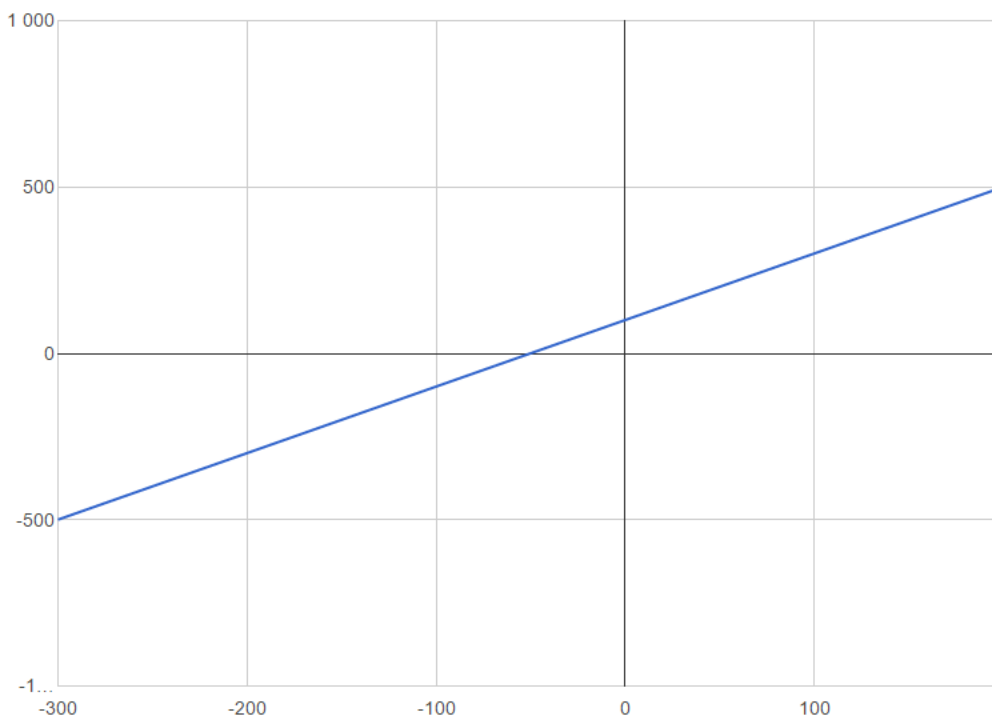


Figura 6.8 - Trajetória para testes de velocidade.

Para o primeiro caso de estudo, o veículo partiu da posição $x_0 = -300$ m e $y_0 = -500$ m, com uma velocidade de movimento de $v_x = 0,5 \text{ ms}^{-1}$ e $v_y = 1 \text{ ms}^{-1}$, que resulta numa velocidade $v = 1,12 \text{ ms}^{-1}$.

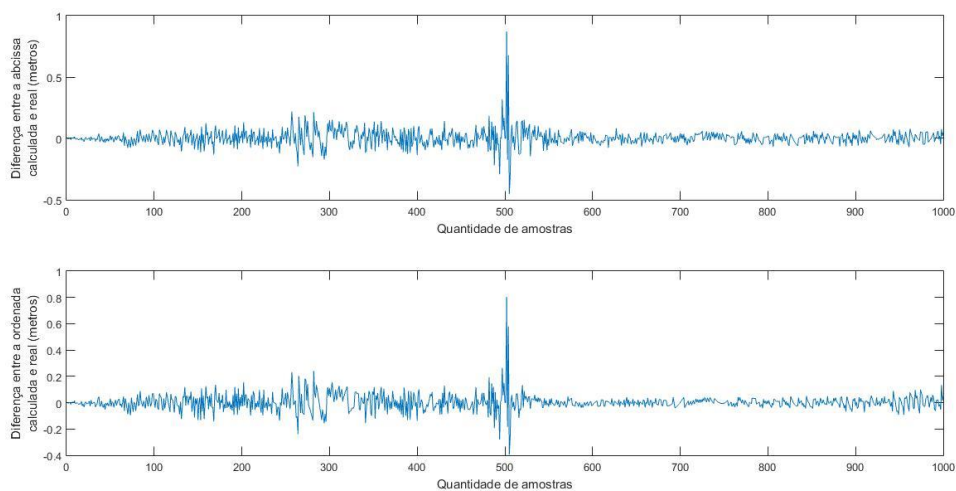


Figura 6.9 - Diferença entre a posição calculada pelo algoritmo e a real.

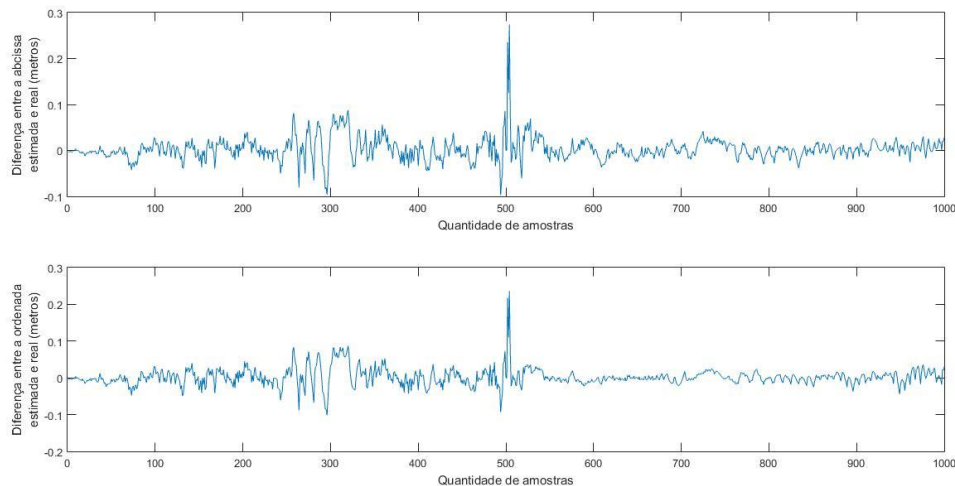


Figura 6.10 - Diferença entre a posição estimada pelo filtro e a real.

Da análise dos resultados do primeiro teste, expressos nas figuras 6.9 e 6.10, entende-se que o erro associado à estimação de posição, é reduzido, praticamente desprezável.

No segundo teste, para manter a trajetória anterior aumentou-se, para o dobro, a velocidade segundo o eixo das abscissas e das ordenadas, ficando com $v_x=1 \text{ ms}^{-1}$ e $v_y=2 \text{ ms}^{-1}$, respetivamente, o que advém uma velocidade de $v= 2,24 \text{ ms}^{-1}$. Para manter o valor máximo atingido por “x” e por “y” diminui-se, para metade, o número de amostras, o que resulta em apenas 500.

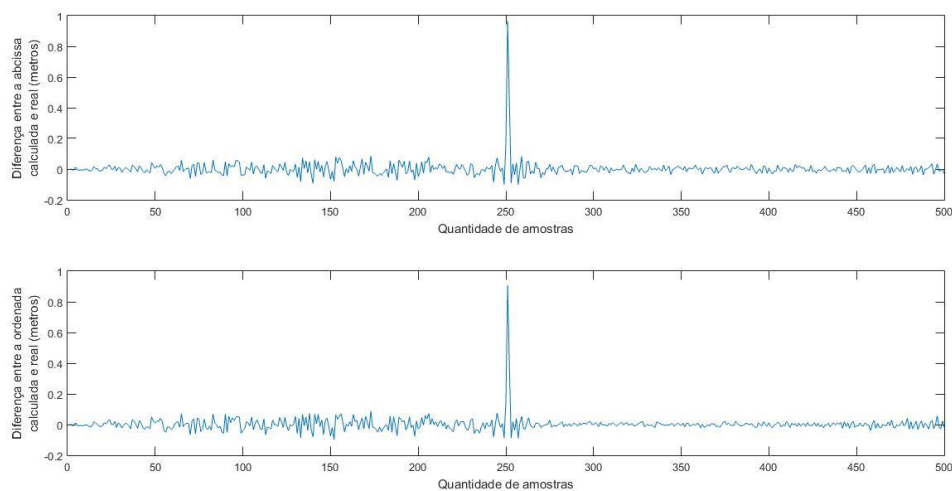


Figura 6.11 - Diferença entre a posição calculada pelo algoritmo e a real.

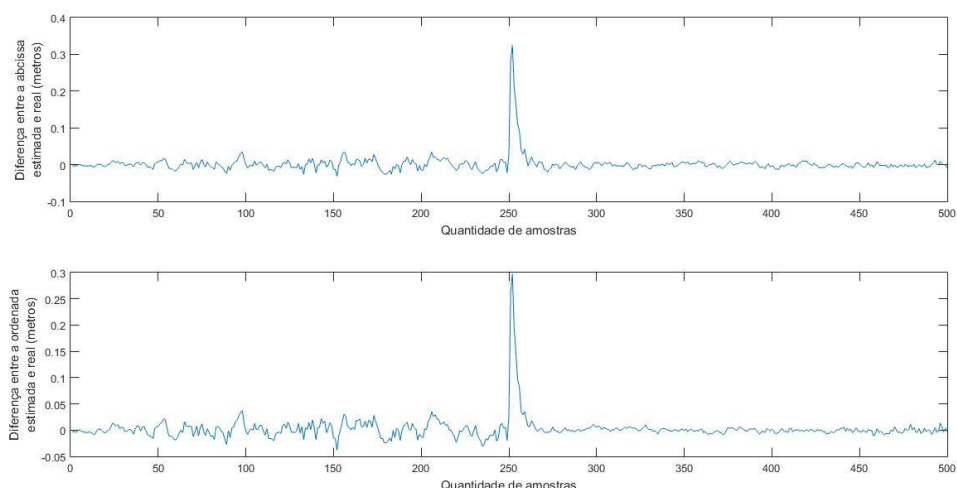


Figura 6.12 - Diferença entre a posição estimada pelo filtro e a real.

Examinando o produto do teste anterior, expresso nas figuras 6.11 e 6.12, nota-se que os erros existentes, em comparação com o caso anterior, são ligeiramente mais reduzidos, logo são desprezáveis.

Tendo em conta os casos de estudo apresentados neste subcapítulo, e através da análise dos resultados anteriores, e da comparação dos dois casos, é possível concluir que a velocidade não é um fator com muita influência nos erros existentes na estimação de posição, pois, quando se aumentou, para o dobro, a velocidade do veículo o ruído presente nos resultados diminuiu, mas essa diminuição não foi algo de muito significativo.

6.3. Testes com dados perdidos e/ou com ruído

Este conjunto de testes servirá para analisar a eficácia do sistema quando sujeito a fatores de erro, como a perda de dados e/ou receber dados com algum ruído. Desta forma, foram introduzidos erros nos dados enviados que, através de um pequeno algoritmo desenvolvido, foi associado uma probabilidade de 5% de perda de dados e outra probabilidade de 5% de ruído, ou seja, se aquando da emissão, uma destas probabilidades ocorra, o sinal acústico, não será emitido ou será emitido com um ruído de 20% o seu valor, respetivamente.

Para este grupo de teste foi utilizado sempre a mesma posição das balizas, $l_{xx}= 500$ m, $l_{yx}= 0$ m e $l_{yy}= 500$ m, e o referencial representado pela figura 6.2, com um conjunto de 600 amostras. O veículo move-se a uma velocidade $v_x=1.8$ ms⁻¹ e $v_y=-1.1$ ms⁻¹, e parte do ponto $x_0=-500$ m e $y_0=500$ m, com a trajetória representada na figura 6.13.

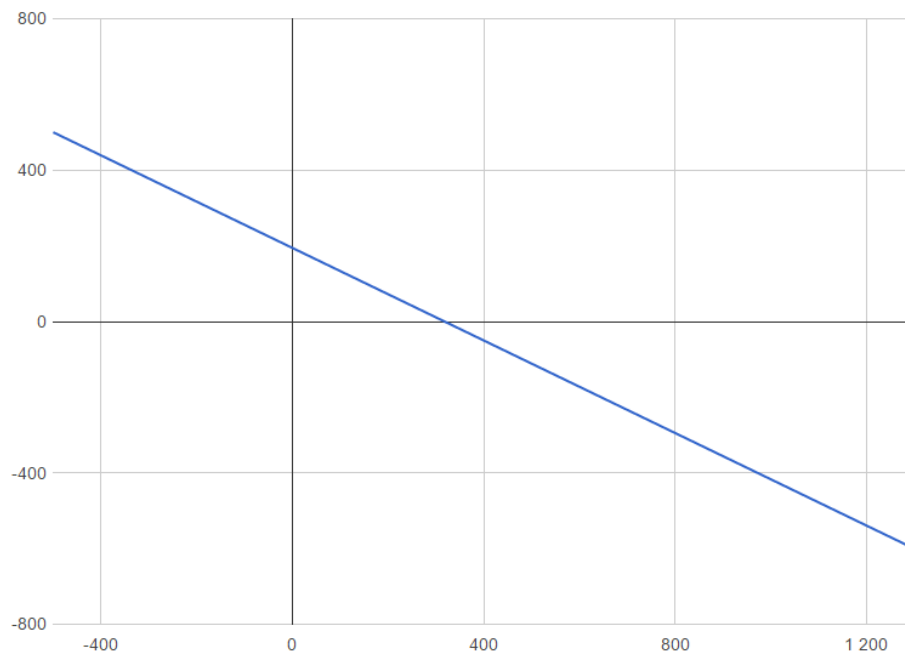


Figura 6.13 - Trajetória para testes com dados perdidos e/ou ruído.

No primeiro teste foi realizado um ensaio com estes valores sem existir perdas nem erros, que serve de controle para os outros testes, com os resultados representados nas figuras 6.14, 6.15 e 6.16.

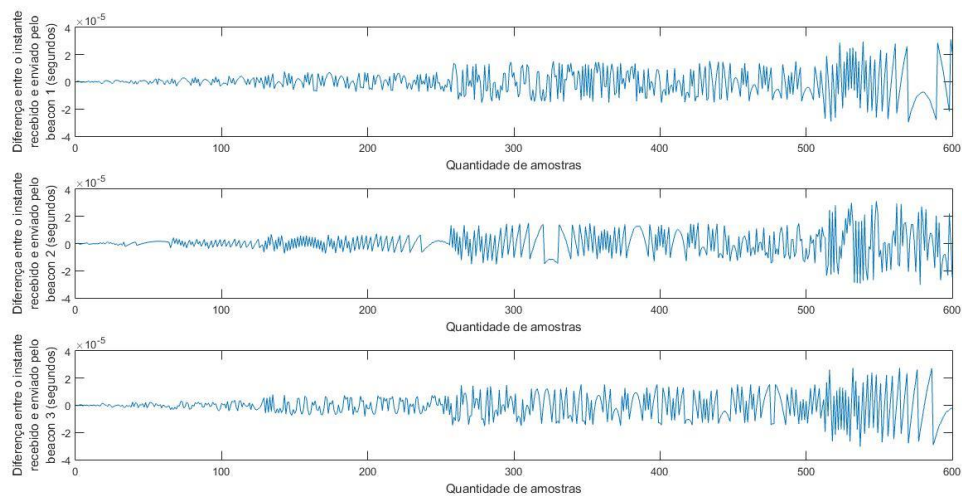


Figura 6.14 - Diferença entre instantes recebido e enviados, usado como controle.

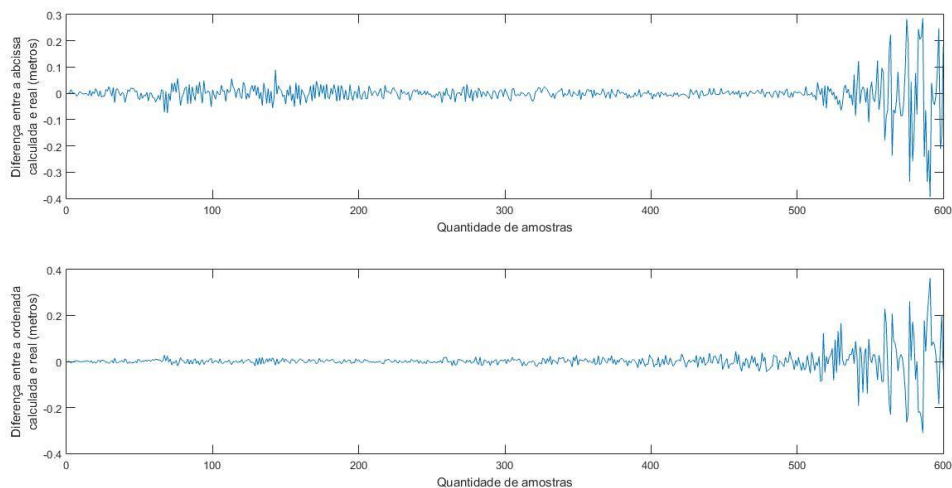


Figura 6.15 - Diferença entre a posição calculada pelo algoritmo e a real, usado como controle.

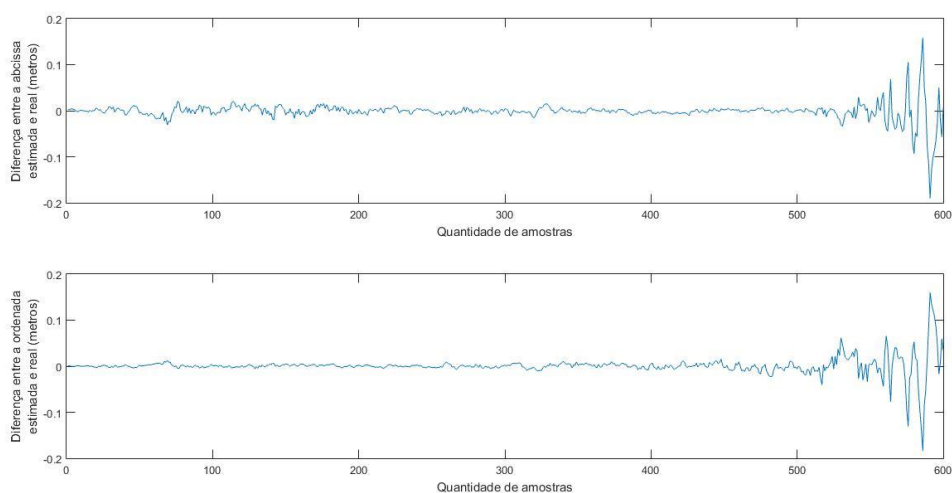


Figura 6.16 - Diferença entre a posição estimada pelo filtro e a real, usado como controle.

No segundo ensaio apenas se submeteu o sistema à probabilidade de falhas de envio, no segundo a apenas ruído nos dados e no terceiro sujeitou-se o sistema a ambos os casos. Desta forma obtiveram-se os resultados expressos na figura 6.17, 6.18 e 6.19.

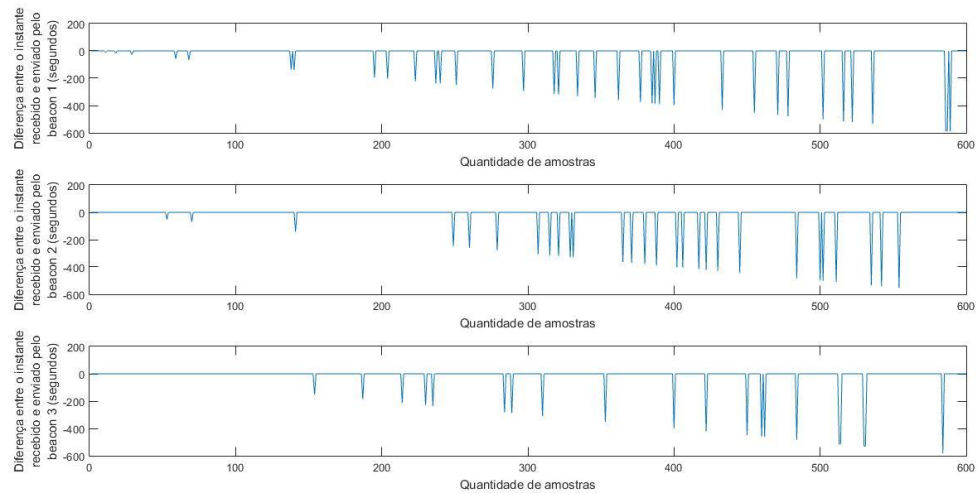


Figura 6.17 - Diferença entre instantes recebido e enviados, com perda de dados.

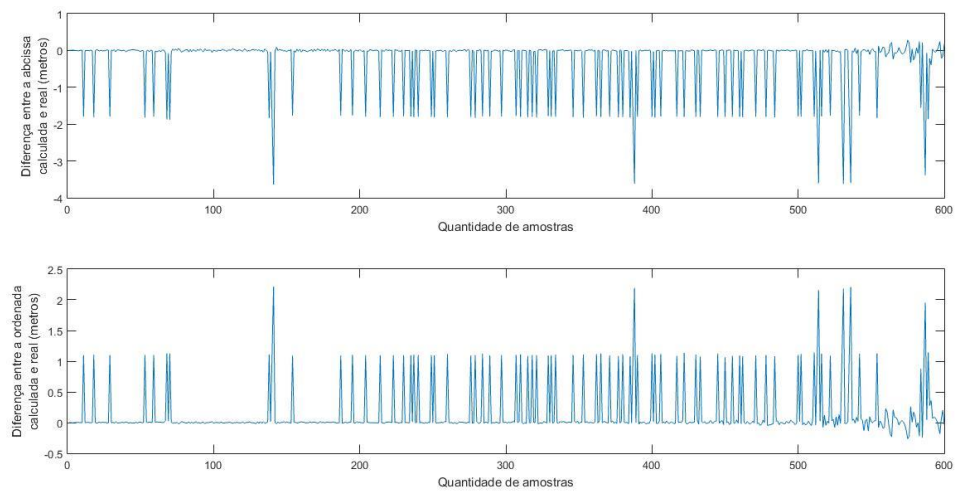


Figura 6.18 - Diferença entre a posição calculada pelo algoritmo e a real, com perda de dados.

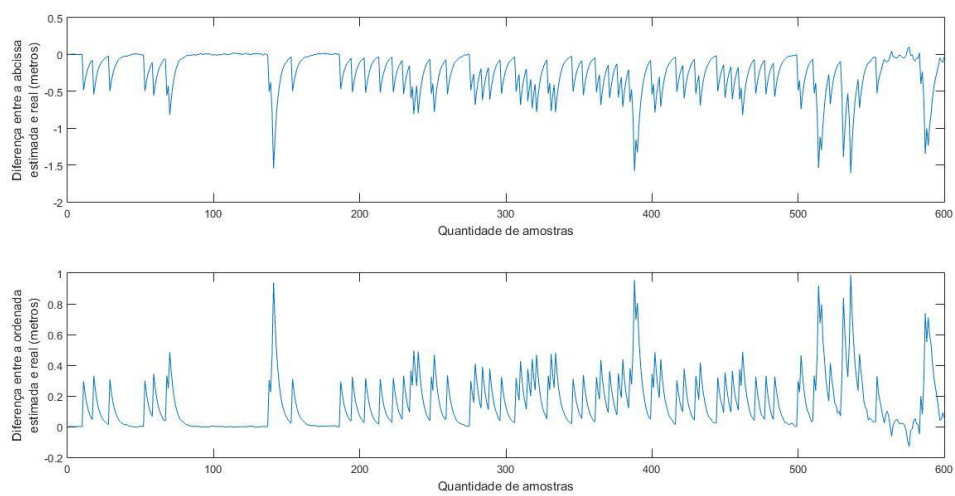


Figura 6.19 - Diferença entre a posição estimada pelo filtro e a real, com perda de dados.

Na figura 6.17, todos os picos negativos presentes no gráfico são os dados que foram não foram enviados, este fenômeno é probabilístico, como mencionado anteriormente, logo o sistema não é capaz nem foi desenhado de forma a combater estas perdas assumindo um valor próximo do que iria ser recebido, em vez disso quando existe uma perda de um sinal o algoritmo assume esse instante como zero, daí as diferenças elevadas e todas negativas. Esta forma de tratar os dados é a mais indicada, pois aquando da análise dos dados recolhidos pelo veículo é possível verificar de uma forma eficaz se ocorreu perda de dados e até mesmo quais deles é que não foram escutados.

Da combinação das figuras 6.17 e 6.18, conclui-se que sempre que existe perda de um sinal, existe grande ruído, isto deve-se a como o algoritmo foi implementado, ou seja, na ausência de um sinal o sistema assume a posição atual igual à posição anterior, como o veículo se está a mover com uma velocidade de $v_x=1.8 \text{ ms}^{-1}$ e $v_y=-1.1 \text{ ms}^{-1}$, é normal que existam erros na ordem dos 2 m e dos -1 m nas coordenadas das abcissas e das ordenadas, respetivamente.

Observando o gráfico 6.19, verifica-se que este também foi afetado pelo ruído presente no cálculo das coordenadas através de apenas o algoritmo matemático, no entanto apresenta uma boa atenuação deste ruído face ao resultado presente na figura 6.18.

No próximo ensaio sujeitou-se o sistema a dados com ruído, como mencionado anteriormente. Neste teste utilizou-se as mesmas condições anteriores, ou seja, mesma velocidade, referencial, posição inicial e posição dos *beacons*.

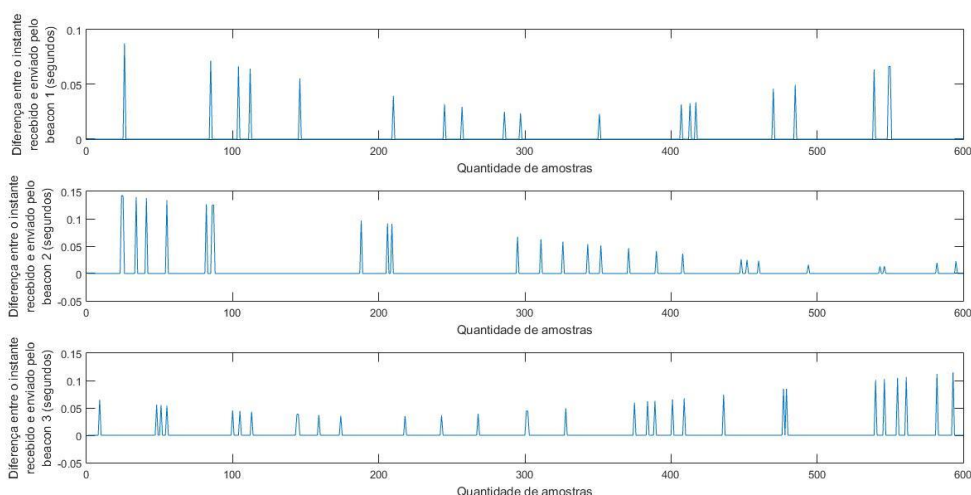


Figura 6.20 - Diferença entre instantes recebido e enviados, com ruído nos dados.

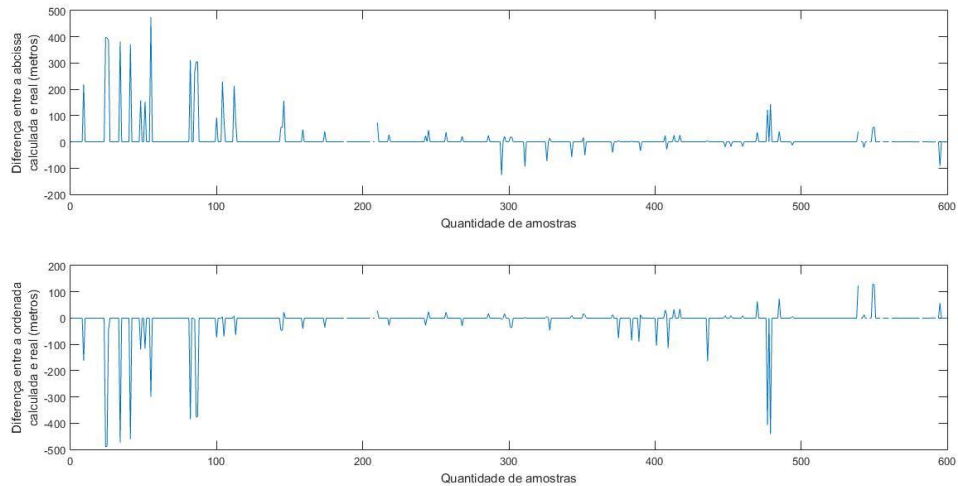


Figura 6.21 - Diferença entre a posição calculada pelo algoritmo e a real, com ruído nos dados.

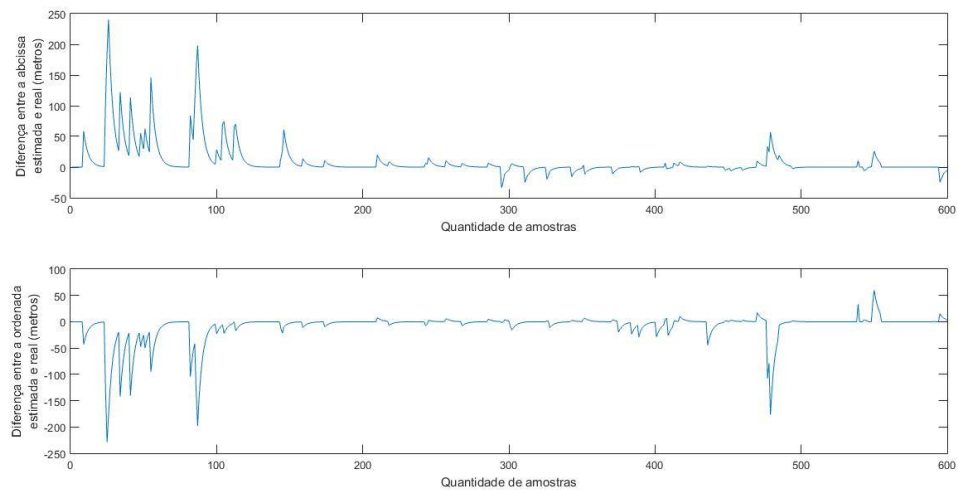


Figura 6.22 - Diferença entre a posição estimada pelo filtro e a real, com ruído nos dados.

Do resultado deste segundo teste, presente nas figuras 6.20, 6.21 e 6.22, é de salientar que os erros existentes nas medições se tornam muito elevados. Este acontecimento é devido ao erro introduzido nos instantes, tendo em conta que quando um instante é emitido com ruído esta medição de posição corresponde a uma coordenada absurda na trajetória. Como o cálculo das coordenadas depende unicamente destes instantes, se estes forem perturbados, inserindo-lhe erros desta ordem, a estimação de posição para esses instantes deveria ser descartada. No entanto analisando os resultados, é possível verificar que mesmo com todo o ruído, o veículo encontra sempre o “caminho certo”.

Neste último teste também foram utilizadas as mesmas condições dos anteriores, apenas com a diferença que neste ensaio, irá ser introduzido ruído nos instantes escutados e também existirão perdas de sinais.

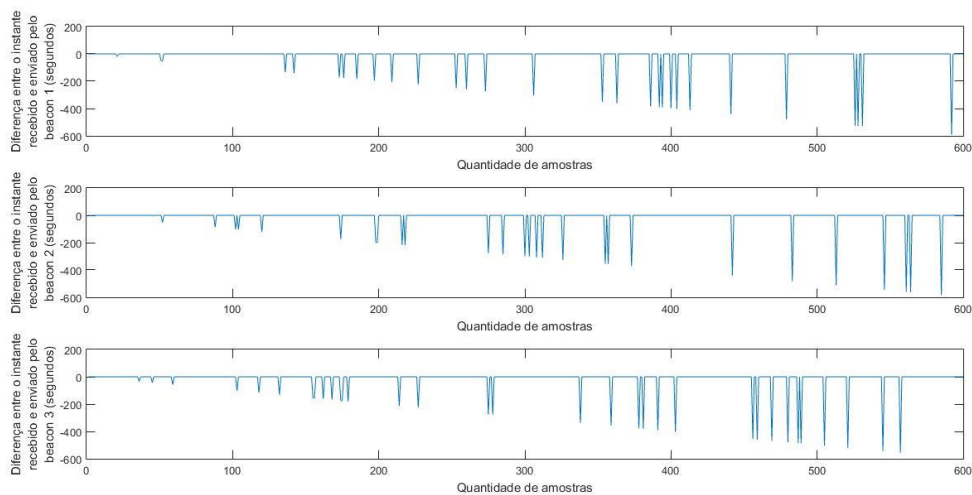


Figura 6.23 - Diferença entre instantes recebido e enviados, com ruído e perda nos dados.

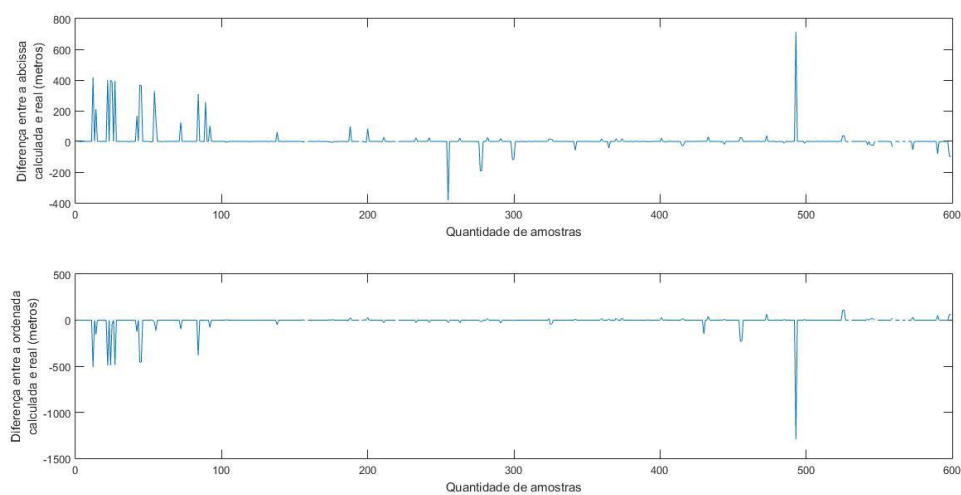


Figura 6.24 - Diferença entre a posição calculada pelo algoritmo e a real, com ruído e perda nos dados.

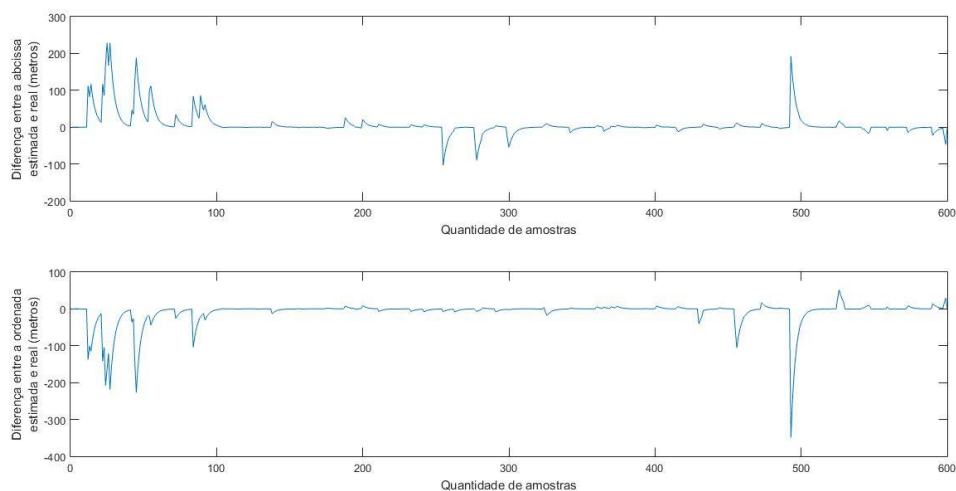


Figura 6.25 - Diferença entre a posição estimada pelo filtro e a real, com ruído e perda nos dados.

Com os resultados obtidos, expressos nas figuras 6.23, 6.24 e 6.25, verifica-se que o erro é da mesma ordem que do teste com apenas ruído nos instantes. Tendo isto em conta é possível afirmar que a maior causa de erros na estimação da posição é o ruído associado aos instantes escutados. Este ruído pode aumentar devido ao ambiente que o veículo está sujeito, ou seja, a sujidade da água, a existência de obstáculos no caminho e até mesmo uma má estimação da velocidade do som na água provoca este tipo de erros. Concluo ainda que o sistema está preparado a perda dos sinais acústicos enviados pelos transdutores, pois os erros que as medições apresentaram são aceitáveis, ao contrário dos erros nas medições com ruído nos dados.

6.4. Testes com trajetórias complexas

Neste grupo de testes irei realizar trajetórias diferentes das anteriores, ou seja, em vez de o veículo se mover apenas numa direção específica irá se movimentar com em várias direções, ou seja, com velocidades diferentes a partir de certos pontos. Desta forma é possível testar a robustez do sistema para trajetórias mais complexas.

No primeiro teste pretende-se efetuar uma trajetória semelhante a um quadrado, para isso, usei o referencial representado na figura 6.2, com $l_{xx} = 500$ m, $l_{yx} = 0$ m e $l_{yy} = 500$ m.

Nesta trajetória, é necessário que o veículo mude de velocidade de movimento pelo menos três vezes, de modo a efetuar quatro linhas que formem um quadrado, para isso dividiu-se o trajeto em quatro partes, representado pela figura 6.26, de modo a ser mais facilmente entendido:

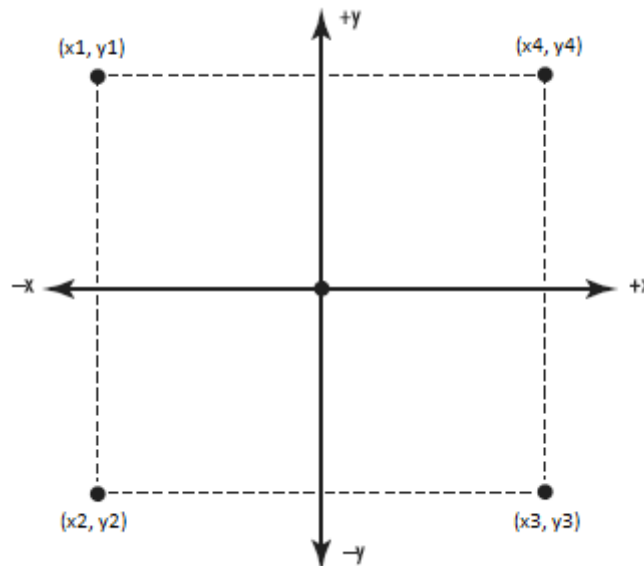


Figura 6.26 - Trajetória pretendida para o veículo.

- Primeira parte - Parte do ponto $x_1 = -300$ m $y_1 = 300$ m com $v_x = 0$ ms⁻¹ e $v_y = -1$ ms⁺¹.
- Segunda parte - Parte do ponto $x_2 = -300$ m $y_2 = -300$ m com $v_x = 1$ ms⁻¹ e $v_y = 0$ ms⁺¹.
- Terceira parte - Parte do ponto $x_3 = 300$ m $y_3 = -300$ m com $v_x = 0$ ms⁻¹ e $v_y = 1$ ms⁺¹.
- Quarta parte - Parte do ponto $x_4 = 300$ m $y_4 = 300$ m com $v_x = -1$ ms⁻¹ e $v_y = 0$ ms⁺¹.

De forma a satisfazer esta rota são necessárias 600 amostras em cada parte. Na figura 2.27, 2.28, 2.29 e 2.30 estão presentes os resultados deste teste.

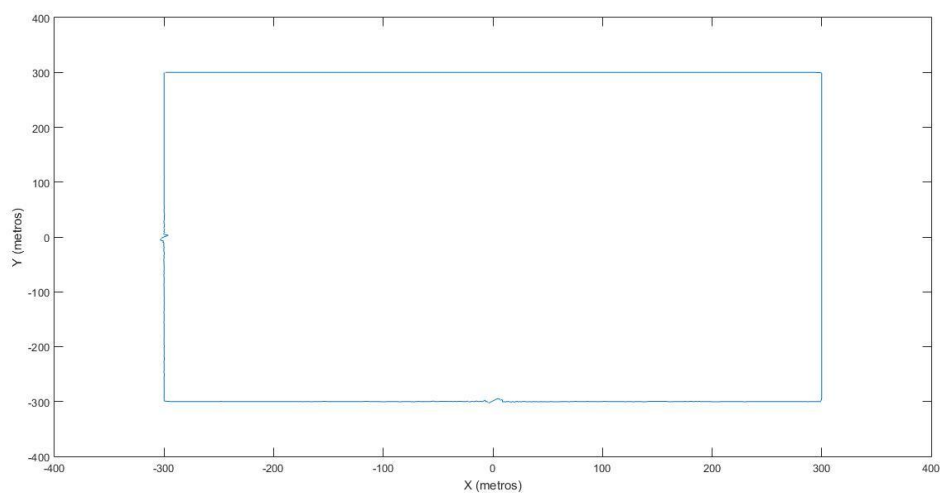


Figura 6.27 - Trajeto do veículo.

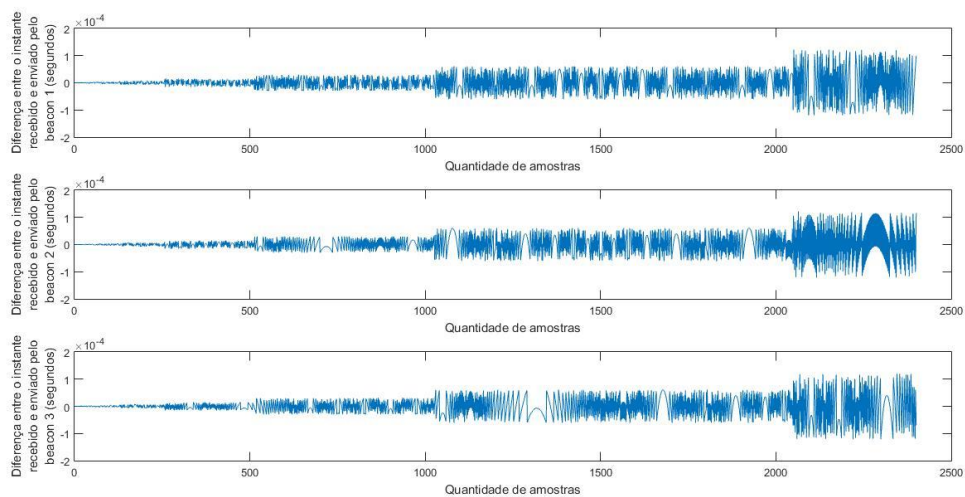


Figura 6.28 - Diferença entre instantes recebido e enviados, para ensaio do quadrado.

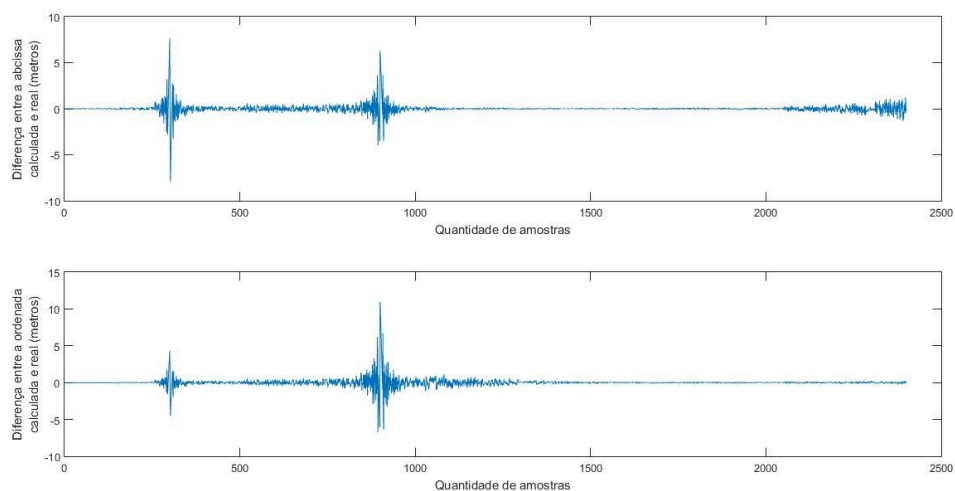


Figura 6.29 - Diferença entre a posição calculada pelo algoritmo e a real, para ensaio do quadrado.

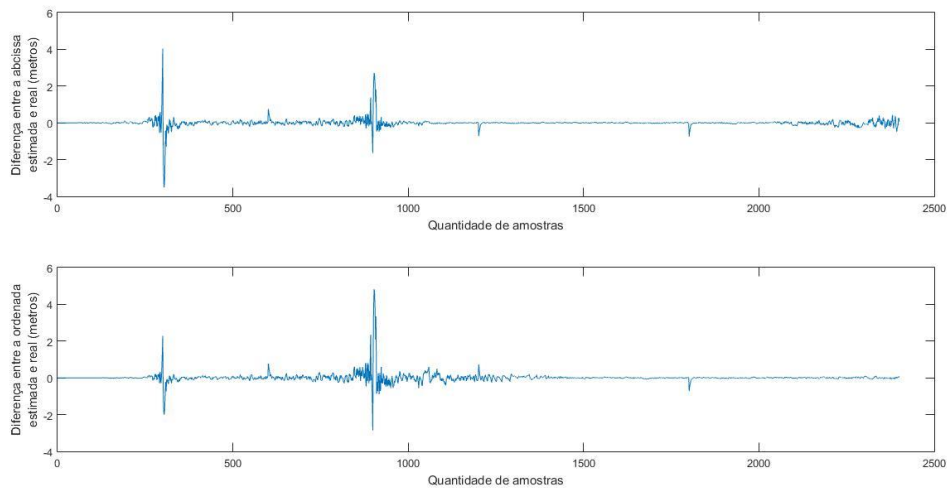


Figura 6.30 - Diferença entre a posição estimada pelo filtro e a real, para ensaio do quadrado.

Analisando o produto deste ensaio, verifica-se que o sistema é capaz de seguir o veículo mesmo com este tipo de mudanças de velocidade e trajetória, apesar de conter alguns erros, o algoritmo segue perfeitamente a trajetória que o veículo faz, como representado em 6.27.

Para o segundo ensaio

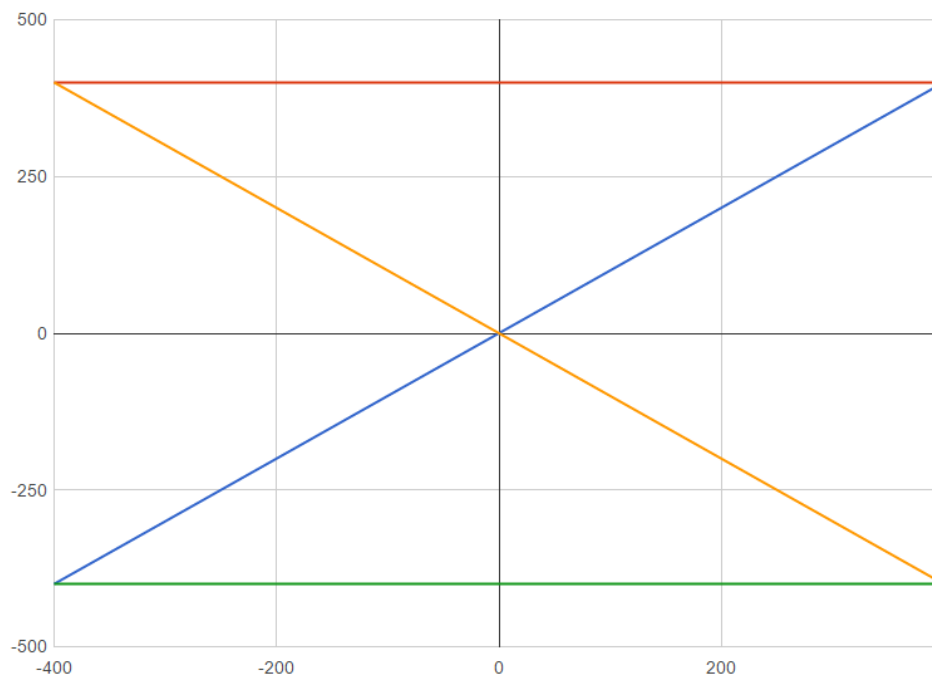


Figura 6.31 - Trajeto pretendido do veículo.

- Primeira parte (azul) - Parte do ponto $x_1 = -400$ m $y_1 = -400$ m com $v_x = 1$ ms⁻¹ e $v_y = 1$ ms⁻¹.
- Segunda parte (vermelho) - Parte do ponto $x_2 = 400$ m $y_2 = 400$ m com $v_x = -1$ ms⁻¹ e $v_y = 0$ ms⁻¹.
- Terceira parte (amarelo) - Parte do ponto $x_3 = -400$ m $y_3 = 400$ m com $v_x = 1$ ms⁻¹ e $v_y = -1$ ms⁻¹.

- Quarta parte (verde) - Parte do ponto $x_4 = 400$ m $y_4 = -400$ m com $v_x = -1$ ms⁻¹ e $v_y = 0$ ms⁻¹.

De forma a satisfazer esta rota são necessárias 800 amostras em cada parte. Na figura 2.32, 2.33, 2.34 e 2.35 estão presentes os resultados deste teste.

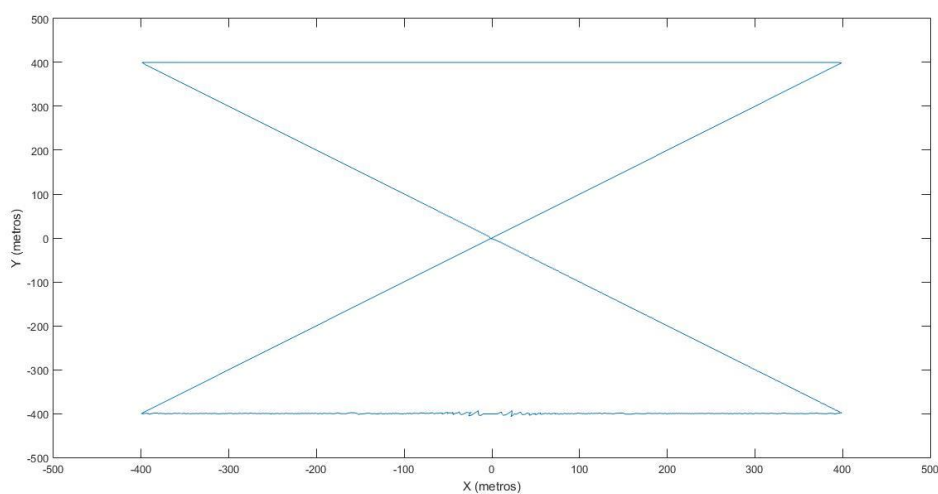


Figura 6.32 - Percurso percorrido pelo veículo.

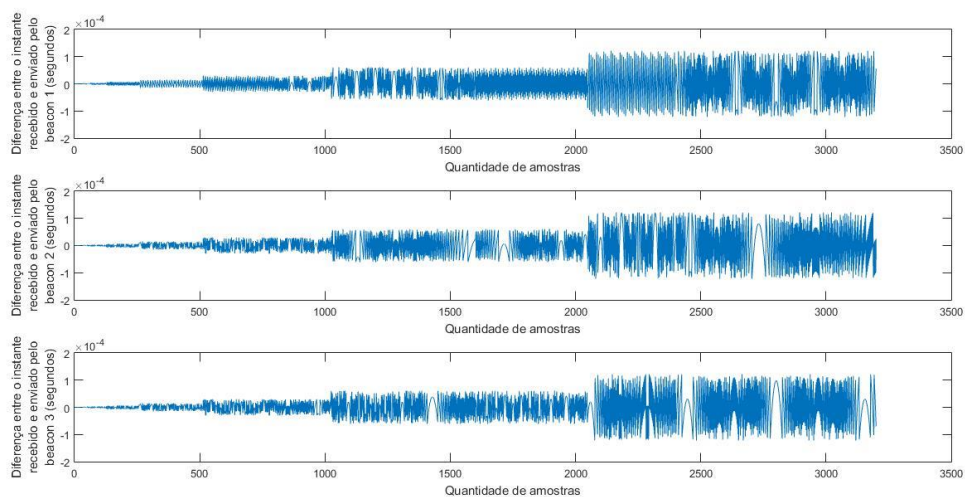


Figura 6.33 - Diferença entre instantes recebido e enviados, para ensaio da cruz.

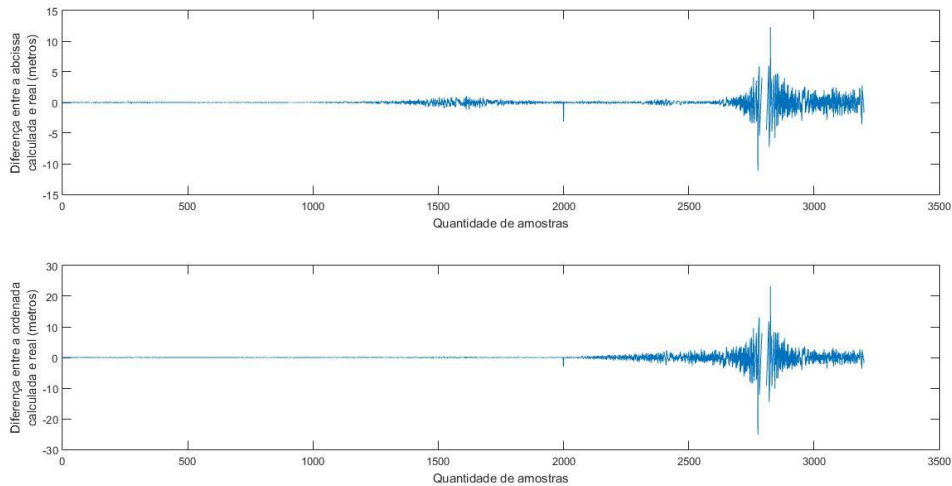


Figura 6.34 - Diferença entre a posição calculada pelo algoritmo e a real, para ensaio da cruz.

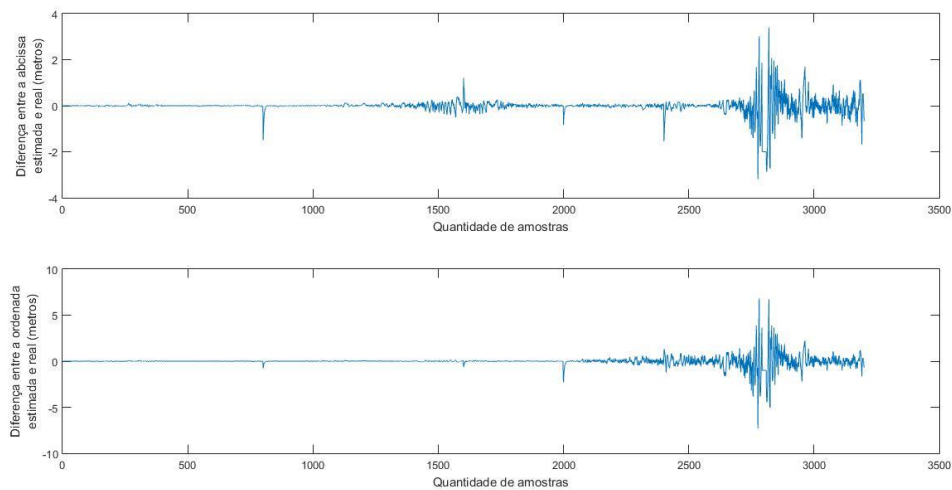


Figura 6.35 - Diferença entre a posição estimada pelo filtro e a real, para ensaio da cruz.

Observando os resultados deste último ensaio, o da cruz, repara-se que existem erros em algumas medidas, no entanto, tal como no caso anterior, o algoritmo segue o veículo de forma estupenda.

Com isto é possível afirmar que o sistema iria funcionar de forma satisfatória em condições reais, e se assumíssemos que existiriam poucos erros nos instantes recebidos, poder-se-ia afirmar que o algoritmo final seguiria de uma forma muito satisfatória um AUV.

Capítulo 7

Conclusão e trabalho futuro

Neste capítulo serão feitas conclusões gerais acerca da elaboração desta dissertação. Será discutida a satisfação face aos objetivos iniciais. Como reflexão do trabalho realizado e do que foi aprendido, são apresentadas propostas de trabalho futuro para continuação e melhoria do que foi feito nesta dissertação.

7.1. Satisfação dos objetivos

O principal objetivo desta dissertação era desenvolver uma ferramenta de navegação acústica passiva de um AUV, baseado em diferenças, que utilizasse os instantes escutados para triangular e desta forma estimar a posição do veículo.

Para este efeito foram desenvolvidos vários algoritmos que processam a informação em tempo real desde que ela é recebida até ser interpretada e representada graficamente. Apesar do ambiente ser simulado, os algoritmos foram desenvolvidos com cuidado para que fossem facilmente adaptáveis para a situação real. Foi tido cuidado para não fazer os algoritmos dependerem dos dados utilizados e foi tida em conta a adaptabilidade de todo o software a outros tipo de missão deste tipo.

Face à implementação destes algoritmos, em relação à aquisição e tratamentos da informação, conclui-se que a aplicação desenvolvida é capaz de processar os dados e computá-los de forma a obter a estimação da posição do veículo em coordenadas cartesianas.

Depois de analisados os resultados, a associação entre os dados obtidos e a trajetória esperada da missão, conclui-se que foi bem sucedida.

Através da velocidade estimada, foi possível desenvolver um filtro de Kalman capaz de filtrar estes dados, de tal forma que diminui o ruído de cada medição em pelo menos metade.

Assim conclui-se que o conjunto dos produtos obtidos constituem uma ferramenta que, a partir da pouca informação que é possível obter do veículo autónomo, produz e analisa informações que permitem a navegação acústica de um veículo.

7.2. Trabalho futuro

A elaboração desta dissertação deixou em aberto possibilidades de melhorias que poderão ser exploradas. Quanto à aquisição de dados, seria uma mais valia recorrer a dados recolhidos de missões anteriores e fazer uma caracterização do ruído a eles associados. Este procedimento permitiria a filtragem dos dados muito mais eficazmente, contribuindo não só para a eficácia da navegação em si, mas para também softwares que utilizem a informação vinda deste sistema para outras aplicações.

Testes em ambiente real seria algo de muito vantajoso para fortalecer o sistema, tornando-o desta forma mais eficiente, para condições reais. Em testes deste tipo, outra mais valia para o sistema seria a utilização de um maior número de balizas acústicas, dado os problemas indicados anteriormente, se existisse a possibilidade de escolha de um vasto conjunto de balizas a estimação da posição do veículo seria muito mais precisa.

No desenvolvimento desta dissertação só se utilizaram trajetórias retilíneas para as missões do veículo. Uma capacidade que seria importante desenvolver seria a de processar trajetórias que não fossem deste cariz.

Referências

- [1] Shawnee K. Vickery, Sonardyne, Inc, "Acoustic Positioning Systems "A practical overview of current systems", em *Dynamic Positioning Conference*, 13 - 14 Outubro, 1998.
- [2] Nuno Cruz, Luís Madureira, Aníbal Matos, e Fernando Lobo Pereira, "A versatile acoustic beacon for navigation and remote tracking of multiple underwater vehicles.," em *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, Novembro 2001.
- [3] Rui Almeida, José Melo e Nuno Cruz, "Characterization of Measurement Errors in a LBL Positioning System," em *OCEANS*, Xangai, China, 10-13 April 2016.
- [4] Nuno A. Cruz e Anibal C. Matos, "The MARES AUV, a Modular Autonomous Robot for Environment Sampling," em *OCEANS 2008*, Setembro 2008.
- [5] Aníbal Matos, Rui Almeida e Nuno Cruz, "Man Portable Acoustic Navigation Buoys," em *OCEANS 2016*, Xangai, Abril 2016.
- [6] Liam Paull, Sajad Saeedi, Mae Seto e Howard Li, "AUV Navigation and Localization: A Review," *IEEE Journal of Oceanic Engineering*, vol. 39, Janeiro 2014.
- [7] Alessio Turetta, Giuseppe Casalino, Enrico Simetti, Alessandro Sperindè, Sandro Torelli, "Analysis of the Accuracy of a LBL-based Underwater Localization Procedure," em *Oceans - St. John's, 2014*, Universidade de Génova, Itália, 2014.
- [8] Aníbal Matos, Nuno Cruz, Alfredo Martins e Fernando Lobo Pereira, "Development and implementation of a low-cost LBL navigation system for an AUV," em *OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century*, Seattle, WA, USA, 13-16 Setembro 1999.
- [9] Bruno M. Ferreira, Aníbal C. Matos, Helder S. Campos, Nuno A. Cruz, "Localization of a sound source: Optimal positioning of sensors carried on autonomous surface vehicles," em *Oceans - San Diego, 2013*, San Diego, CA, USA, 23-27 Setembro 2013.
- [10] Brian Bingham, "Predicting the Navigation Performance of Underwater Vehicles," em *International Conference on Intelligent Robots and Systems*, St. Louis, USA, Outubro 11-15, 2009.

- [11] Ryan M. Eustice, Louis L. Whitcomb, Hanumant Singh e Matthew Grund, “Recent Advances in Synchronous-Clock One-Way-Travel-Time Acoustic Navigation,” em *OCEANS 2006*, Boston, MA, USA, 18-21 Setembro 2006.
- [12] Ryan M. Eustice, Hanumant Singh e Louis L. Whitcomb, “Synchronous-Clock One-Way-Travel-Time Acoustic Navigation for Underwater Vehicles,” *Journal of Field Robotics*, Setembro 2010.
- [13] Donald Thomson, “Acoustic Positioning Systems,” em *For The Hydrographic Society, Hydrofest*, Escócia, 2005.
- [14] Andrea Caiti, Vincenzo Calabrò, Tommaso Fabbri, Davide Fenucci e Andrea Munafò, “Underwater communication and distributed localization of AUV teams,” em *OCEANS*, Xangai, China, 10-13 April 2016.
- [15] Bruno Ferreira, Aníbal Matos, Nuno Cruz, “Optimal positioning of autonomous marine vehicles for underwater acoustic source localization using TOA measurements,” em *Underwater Technology Symposium (UT), 2013 IEEE International*, Tóqui, Japão, 5-8 Março 2013.